REPORT FROM THE FIRST WORKSHOP ON EXTREMELY LARGE DATABASES

***J Becla*[*1] *and K-T Lim*[2]**

Stanford Linear Accelerator Center, Menlo Park, CA 94025, USA
[*1] *Email: becla@slac.stanford.edu*
[2] *Email: ktl@slac.stanford.edu*

ABSTRACT

Industrial and scientific datasets have been growing enormously in size and complexity in recent years. The largest transactional databases and data warehouses can no longer be hosted cost-effectively in off-the-shelf commercial database management systems. There are other forums for discussing databases and data warehouses, but they typically deal with problems occurring at smaller scales and do not always focus on practical solutions or influencing DBMS vendors. Given the relatively small (but highly influential and growing) number of users with these databases and the relatively small number of opportunities to exchange practical information related to DBMSes at extremely large scale, a workshop on extremely large databases was organized. This paper is the final report of the discussions and activities at the workshop.

Keywords: Database, XLDB

1 EXECUTIVE SUMMARY

The workshop was organized to provide a forum for discussions focused on issues pertaining to extremely large databases. Participants represented a broad range of scientific and industrial database-intensive applications, DBMS vendors, and academia.

The vast majority of discussed systems ranged from hundreds of terabytes to tens of petabytes, and yet still most of the potentially valuable data was discarded due to scalability limits and prohibitive costs. It appears that industrial data warehouses have significantly surpassed science in sheer data volume.

Substantial commonalities were observed within and between the scientific and industrial communities in the use of extremely large databases. These included requirements for pattern discovery, multidimensional aggregation, unpredictable query load, and a procedural language to express complex analyses. The main differences were the availability requirements (very high in industry), data distribution complexity (greater in science due to large collaborations), project longevity (decades in science vs. quarter-to-quarter pace in industry) and use of compression (industry compresses and science doesn't). Both communities are moving towards parallel, shared-nothing architectures on large clusters of commodity hardware, with the map/reduce paradigm as the leading processing model. Overall, it was agreed that both industry and science are increasingly data-intensive and thus are pushing the limits of databases, with industry leading the scale and science leading the complexity of data analysis.

Some non-technical roadblocks discussed included funding problems and disconnects between vendors and users, within the science community, and between academia and science. Computing in science is seriously under-funded: the scientific community is trying to solve problems of scale and complexity similar to industrial problems, but with much smaller teams. Database research is under-funded too. Investments by RDBMS vendors in providing scalable multi-petabyte solutions have not yet produced concrete results. Science rebuilds rather than reuses software and has not yet come up with a set of common requirements. It was agreed there is great potential for the academic, industry, science, and vendor communities to work together in the field of extremely large database technology once the funding and sociological issues are at least partly overcome.

Major trends in large database systems and expectations for the future were discussed. The gap between the system sizes desired by users and those supported cost-effectively by the leading database vendors is widening. Extremely

large database users are moving towards the use of solutions incorporating lightweight, flexible, specialized components with open interfaces that can be easily mixed and matched with low-cost, commodity hardware. The existing monolithic RDBMS systems face a potential redesign to move in this direction. Structured and unstructured data are coming together; the textbook approach assuming a perfect schema and clean data is inadequate. The map/reduce paradigm popular in many places lacks efficient join algorithms, and therefore it is likely not the ultimate solution. Recent hardware trends will disrupt database technology, in particular the increasing gap between CPU and I/O capability, and emerging solid-state technologies.

Next steps were discussed. It was agreed that collaborating would be very useful. There should be follow-up workshops, and perhaps smaller working groups should be set up. Defining a standard benchmark focused on data-intensive queries and sharing infrastructure ranging from testbed environments to a wiki for publishing information were also highly desired.

2 About the Workshop

The Extremely Large Database (XLDB) Workshop was organized to provide a forum for discussions focused specifically on issues pertaining to extremely large databases. It was held at SLAC[1] on October 25, 2007. The main goals were to:

- identify trends and major roadblocks related to building extremely large databases,
- bridge the gap between users trying to build extremely large databases and database vendors,
- understand if and how open source projects such as the LSST Database can contribute to the previous two goals in the next few years.

The workshop's website can be found at: http://www-conf.slac.stanford.edu/xldb07.

The agenda is reproduced in Appendix A.

The workshop organizing committee was composed of Jacek Becla (chair), Kian-Tat Lim, Andrew Hanushevsky and Richard Mount.

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 insure an appropriate balance between participants from each community. Based on feedback from participants and the workshop outcome, this strategy turned out to be very successful.

The workshop was attended by 55 people from industry (database users and vendors), the scientific community (database users) and academia (database research). During the panel discussions, the XLDB user community from industry was represented by AOL, AT&T, EBay, Google and Yahoo!. The database vendors present included Greenplum, IBM, Microsoft, MySQL, Netezza, Objectivity, Oracle, Teradata and Vertica. Academia was represented by Prof. David DeWitt from the University of Wisconsin and Prof. Michael Stonebraker from M.I.T. The scientific community had representation from CERN, the Institute for Astronomy at the University of Hawaii, IPAC, George Mason University, JHU, LLNL, LSST Corp., NCSA, ORNL, PNL, SDSS, SLAC, U.C. Davis and U.C. Santa Cruz.

The user group was selected to represent a broad range of database applications. On the industrial side, these ranged from search engines (Google, Yahoo!) and web portals (AOL) through on-line bidding systems (EBay) and telecom (AT&T). On the scientific side these ranged from high energy physics (LHC, BaBar) and astronomy (SDSS, PanSTARRS, LSST, 2MASS) through complex biological systems. The names and affiliations of all the attendees can be found in Appendix B.

[1] See the Glossary for definitions of abbreviations used in this document.

2.2 Structure

The workshop was held on a single day to facilitate attendance and encourage constructive discussion. The bulk of the time was spent in highly interactive panel sessions, although there were two initial presentations from the two largest scientific projects: LHC representing today's usage and high energy physics and LSST representing future usage and astronomy. The agenda was divided into three parts:

- user panels from the scientific and industrial communities that were meant to reveal how extremely large databases are used, how they have been implemented, and how the users would like to use them
- vendor and academic responses to the issues heard
- discussion about the future and possible next steps.

2.3 About This Report

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

Section 3 shows how extremely large database systems are used in production now and discusses current technological solutions and problems. Section 4 describes issues related to collaboration among the various groups interested in XLDB. Section 5 summarizes participants' thoughts on the future evolution of XLDB. Section 6 documents the consensus on steps that should be taken next.

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. The Facts in Appendix C give some specifics.

3 TODAY'S SOLUTIONS

This chapter describes the current state of extremely large database technology and practice in both science and industry, as revealed through the panels and discussions at the workshop.

3.1 Scale

The workshop was intended to discuss extremely large databases. Unsurprisingly, the vast majority of the systems discussed have database components over 100 terabytes in size, with 20% of the scientific systems larger than 1 petabyte. All of the industry representatives had more than 10 petabytes of data, and their largest individual systems are all at least 1 petabyte in size.

Size is measured in more than just bytes, however. Industry systems already contain single tables with more than a trillion rows. Science has tables with tens of billions of rows today; multi-trillion-row tables will be required in less than ten years.

Peak ingest rates range as high as one billion rows per hour, with billions of rows a day common.

All users said that even though their databases are already growing rapidly, they would store even more data in databases if it were affordable. Estimates of the potential ranged from ten to one hundred times current usage. The participants unanimously agreed that "*no vendor meets our database needs*".

3.2 Usage

The largest databases described are variations on the traditional data warehouse used for analytics. Common characteristics include a write-once, read-many model; no need for transactions; and the need for simultaneous load and query to provide low-latency access to valuable fresh data. These analytical systems are typically separate from the operational, often on-line transaction processing (OLTP), systems that produce the data.

Representatives from both science and industry described highly unpredictable query loads, with up to 90% of queries being new. An incisive phrase used to describe this was "*design for the unknown query*". Nevertheless, there are common characteristics here, too. Most of the load involves summary or aggregative queries spanning large fractions of the database. Multidimensional queries involving value or range predicates on varying subsets of a large set of attributes are common. While science has been able to develop specialized indexes in some cases, the lack of good general index support for these queries leads to the frequent use of full table scans. To deal with the variability, it was suggested that well-designed systems would monitor their own usage and allow adaptation to the current query load.

As mentioned above, not all the data required by the various projects represented are going into databases today. Some are being discarded, but other data that are less queriable and do not require other database features such as transactions are being managed in alternative, cheaper, more scalable ways like file systems. Most scientific users and a few industrial users store aggregate data in a database while leaving the detailed information outside.

There are other uses of databases, of course. Critical transaction-oriented data management needs are almost universally handled by databases, typically using an off-the-shelf RDBMS. Operational data stores requiring low latency and high availability but with strictly defined query sets were also described. These other uses tend not to be the very largest databases (although they might be the metadata for the largest custom-designed databases).

These many uses often require the use of multiple database packages, with each used in its area of expertise. For example, an off-the-shelf RDBMS might be used as the source of data for a custom map/reduce analysis system, or, inverting the flow, an off-the-shelf RDBMS might be used to hold quick-access aggregates from a different system.

3.3 Hardware and Compression

In order to achieve the scalability necessary to handle these extremely large database sizes, parallelism is essential. I/O throughput was often mentioned as being more important than raw CPU power, although both groups, especially science, have some complex CPU-intensive processing. Most systems use a shared-nothing architecture in which the data was typically partitioned horizontally by row. The total number of nodes used in parallel in a single cluster range as high as tens of thousands for industry, thousands for science.

Increasing the number of nodes exponentially increases the chance of failure. In fact, hardware failures are so common in large systems that they must be treated as a normal case. Today's predominant solution for these clusters is to handle hardware failures through software, rather than relying on high-end hardware to provide high availability. Once software-based, transparent failure recovery is put into place, modestly greater failure probabilities (as high as 4-7% per year for disks in one report) do not affect the overall system availability. Many projects in both science and industry are thus able to use low-end commodity hardware instead of traditional "big iron" shared memory servers. This approach is also typically associated with the use of local disks, rather than network-attached storage or SAN. The number of disk drive spindles is seen as being more important than the total amount of storage, dramatically so for random-access systems. One of the advantages of map/reduce type systems is that they decrease the need for random access.

These large systems also require a large amount of power, cooling, and floor space; the database component of the system was cited as often being one of the largest contributors, as disk arrays can produce large amounts of heat.

One way of conserving both disk space and I/O bandwidth is to compress the data. Everyone in industry is compressing their data in some fashion. The scientific projects, on the other hand, do not typically compress their data, as the structure and composition of the data apparently do not permit compression ratios sufficient to justify the CPU required.

3.4 SQL and the Relational Model

The relational model and the SQL query language have been very successful since the "Great Debate" many years ago. Keeping the data and processing models simple has given them great power to be used in many diverse situations. Many users felt constricted by the implementations available in off-the-shelf RDBMS packages, however.

Often industrial data originates in fully-normalized OLTP systems but then is extracted into denormalized or even semi-structured systems for analysis; it is rare for scientific data (but not metadata) to ever be stored in fully-normalized form. The poor performance of general-purpose billion-to-trillion-row join operations has led users to pre-join data for analysis or use the map/reduce paradigm[1] for handling simple, well-distributed joins.

With full table scans being common, as mentioned above, indexing is of lower priority, although the partitioning scheme is inherently a first-level index. Column-oriented stores are playing an increasing role in these large systems because they can significantly reduce I/O requirements for typical queries.

Procedural languages are required by both science and industry for processing and analyzing data. In industry, higher-level languages such as Sawzall[2], Pig[3], or the Ab Initio ETL[4] tool were mentioned. In science, lower-level languages such as C++ tend to be used.

When industry does use SQL for analytics, the queries are most often generated by tools, rather than hand-coded, with one project reporting as many as 90% of queries being tool-driven. In science, hand-coded queries are common, with even lower-level programmer-level access frequent, as query-generating tools have been tried and generally found insufficient.

Object-relational adapters and object-oriented databases are not felt to be appropriate, at the current level of development, for these extremely large databases, particularly the warehouse-style ones.

Overcoming the barriers to mapping scientific data into the relational model or developing useful new abstractions for scientific data and processing models are seen as necessities to allow scaling of both hardware and people as new extremely large databases are built. Jim Gray worked hard to overcome this barrier, successfully showing in several cases that science can use relational systems.

3.5 Operations

Industrial systems typically require high availability, even during loading and backup. These systems are often integrated into essential business processes. Science, on the other hand, can often deal with availabilities as low as 98%, although not if the downtime interrupts long-running complex queries.

Smaller databases that are components of these systems may require more than just high availability: they may also need to provide real-time or near-real-time response. Examples occur in both science, where detector output must be captured, and industry, where immediate feedback can translate directly into revenue. One industrial project processes as much as 25 terabytes of streaming data per day. On the other hand, some scientific databases may be released as infrequently as once per year, preceded by reprocessing with the latest algorithms and intensive quality assurance processes.

Manageability of these systems was deemed important. Neither group can afford to have a large number of database administrators, or to scale that number with the size of the system.

Replication of data and distribution of systems across multiple sites, sometimes on a worldwide basis, is necessary for both groups to maintain availability and performance, but it does add to the manageability headaches. Science is particularly impacted here, as its collaborations often involve tens to hundreds of autonomous partners around the globe operating in radically different hardware, software, network and cultural environments with varying levels of expertise, while industry can exert firmer control over configurations.

[1] http://labs.google.com/papers/mapreduce-osdi04.pdf
[2] http://labs.google.com/papers/sawzall-sciprog.pdf
[3] http://incubator.apache.org/pig/
[4] http://www.abinitio.com

3.6 Software

While industry may have more resources than science, both do not want to pay more than necessary for database solutions. Both groups often use free and/or open source software such as Linux, MySQL, and PostgreSQL extensively to reduce costs. Both groups also write custom software, although at different levels. Industry tends to implement custom scalable infrastructure, including map/reduce frameworks and column-oriented databases, which provides abstraction for programmers and eventually analysts. Science tends to implement custom top-to-bottom analysis, utilizing minimal data access layers that nevertheless provide isolation from underlying storage to enable implementation on environments that may be heterogeneous across varying locations and across the lifetime of long-running projects.

Stereotypically, industry is fast-moving, operating on a quarter-to-quarter or month-to-month basis, while science moves at a slower pace with decade-long projects. In actuality, industry needs to amortize infrastructure development over multi-year timeframes, while science needs to plan to upgrade or replace technology in mid-project. Software in both cases is continually evolving with new requirements and features.

All participants expressed a need for performing substantial computation on data in databases, not just retrieving it. Extracting patterns from large amounts of data and finding anomalies in such data sets are key drivers for both groups. These analytic, discovery-oriented tasks require near-interactive responses as hypotheses must be repeatedly tested, refined, and verified. The algorithms required to generate useful attributes from scientific data today are often orders of magnitude more computationally expensive, particularly in floating point operations, than those found in industry. As a result, science tends to do more precomputation of attributes and aggregates than industry, though this slows down the discovery process.

The map/reduce paradigm has built substantial mind-share thanks to its relatively simple processing model, easy scalability, and fault tolerance. It fits well with the aforementioned need for full table scans. It was pointed out that the join capabilities of this model are limited, with sort/merge being the primary large-scale method being used today.

3.7 Conclusions

There are substantial commonalities within and between the scientific and industrial communities in the use of extremely large databases. Science has always produced large volumes of data, but contemporary science in many different fields of study is becoming increasingly data-intensive, with greater needs for processing, searching, and analyzing these large data sets. Databases have been playing an increasing role as a result. Industrial data warehouses have surpassed science in sheer volume of data, perhaps by as much as a factor of ten.

The types of queries used by industry and science also exhibit similarities, with multidimensional aggregation and pattern discovery common to both. The overall complexity of business analyses is still catching up to that of science, however.

The relational model is still relevant for organizing these extremely large databases, although industry is stretching it and science is struggling to fit its complex data structures into it.

Both communities are moving towards parallel, shared-nothing architectures on large clusters of commodity hardware, with the map/reduce paradigm as the leading processing model.

4 COLLABORATION ISSUES

This chapter summarizes non-technological roadblocks and problems hindering the building of extremely large database systems, including sociological and funding problems, as revealed through the panels and discussions at the workshop. In some cases possible solutions were discussed.

4.1 Vendor/User Disconnects

The scale of databases required in both science and industry has been increasing faster than even the best RDBMS vendors can handle. Vendors have been working with large customers, learning how best to apply existing technologies to the extremely large scale and discovering requirements for the future, but they are still generally seen as not keeping up. The overall feeling was that they may be building on lagging indicators, instead of leading, resulting in their creating solutions for yesterday's problems. The perception of users was that "*existing RDBMSes could now easily solve all the problems we had five years ago, but not today's problems*". One possible explanation that was mentioned is insufficient exposure of the database vendors to real-world, large scale production problems; there was even a suggestion that vendor representatives should rotate through customer sites to get a better feel for how they operate.

4.2 Internal Science Disconnects

A common perception outside of the scientific community is that that community is inefficient at software development: it rebuilds rather than reuses. One reason cited for this inefficiency was the "free labor" provided by graduate students and post-docs; treating this labor as zero-cost means that sharing code provides little value. On the other hand, significant portions of scientific software cannot be reused, even within the same science, as it performs specialized computations that are tightly coupled with experimental hardware, such as calibration and reconstruction.

The longevity of large scientific projects, typically measured in decades, forces scientists to introduce extra layers in order to isolate different components and ease often unavoidable migrations, adding to system complexity. Unfortunately, those layers are typically used only to abstract the storage model and not the processing model.

In conclusion, the scientific community needs to try harder to agree on common needs, write more efficient software and build more sharable infrastructure to the extent possible.

4.3 Academia/Science Disconnects

In the past, computer scientists working in the database area have tried to collaborate with scientists. These efforts generally failed. Technical failures included difficulties with supporting arrays, lineage, and uncertainty in databases. Social failures included mismatched expectations between the two groups, with computer science able to produce prototypes while science was anticipating production-quality systems. This resulted in a lack of adoption of new techniques and consequently a lack of feedback on the usefulness of those techniques. Jim Gray was much lauded as the exceptional person who managed to span the divide between the two communities by dint of full-time work and his ability to leverage the resources of Microsoft.

A suggestion to have computer science graduate students work in science labs was deemed infeasible. Science project timescales are too long to enable the students to have sufficient publications to provide good career prospects.

It is possible for the scientific community to reengage academia. Both sides must be willing to partner and set reasonable expectations. Science must take data management and databases seriously, not treat them as an afterthought. The primary need is for the community to develop a set of distilled requirements, including building blocks such as desired data types and operations. Projects might contribute key data sets to a data center where academics could access and experiment with them.

4.4 Funding Problems

The high-end commercial systems are very expensive to purchase and to operate. Science certainly cannot afford commercial systems, and even industry has balked at the price tags. Industry has responded by investing in building custom database systems that are much more cost-effective, even though the cost of development can only be amortized across internal customers. Science, meanwhile, has underinvested in software development. With problems of similar scale and complexity to industry, the scientific community is trying to get by with much smaller teams. While industry does need to move faster and thus has shorter development timescales, its return-on-investment timescales are similarly shortened.

Database research within computer science was felt to be underfunded as well. The result has been a lack of major changes and discoveries in the database field since the introduction of the relational model. One pithy commenter said, "*twenty years of research, and here we go, we have map/reduce*."

4.5 Conclusions

There is great potential for the academic, industry, science and vendor communities to work together in the field of extremely large database technology. Past difficulties that have limited progress in this area must be overcome, and increased funding for database research and scientific infrastructure must be obtained.

5 THE FUTURE OF XLDB

This chapter describes trends in database systems and expectations for the future. These were the subject of several discussions, primarily during the vendor and academic panels.

5.1 State of the Database Market

Based on the discussions at the workshop, standard RDBMS technology is not meeting the needs of extremely large database users. The established database vendors do not seem to be reacting quickly enough to the new scale by adapting their products. It appears instead that the gap between the system sizes they support cost-effectively and those desired by users is widening.

Meanwhile, open source general purpose RDBMS software is becoming more capable and gaining in performance, at a low price point. Some of these systems include open interfaces that allow users to plug customized components such as storage engines into a standard, well-tested framework to tune the product to their needs. The open source database community has not yet solved the scalability problems either, but users willing to invest in custom development have found this software useful as components of larger scalable systems.

Simultaneously, specialized niche engines, including object-oriented, columnar, and other query-intensive, non-OLTP databases, are increasingly finding traction on the large-scale end. Order of magnitude improvements in minimizing I/O through compression and efficient clustering of data, effortless scalability, and the resulting gains in the ratio of performance to price make the deployment of these systems worthwhile, despite their unique nature and difficulties with interoperability.

The traditional RDBMS vendors are facing increased competition because of these trends. One participant provocatively suggested that as a result, they will fade away in the next ten to twenty years. Even if that proves not to be the case, they will likely have to undergo a substantial redesign to succeed in the extremely large database market.

People managing very large data sets strongly dislike monolithic systems. Such systems are inflexible, difficult to scale and debug, and tend to lock the customer into one type of hardware, frequently the high-end, expensive type. The emerging trend is to build data management systems from specialized, lightweight components, mixing and matching these components with an appropriate mixture of low-cost, commodity hardware (CPU, memory, flash, fast disks, slow disks) to achieve the ultimate balance.

Structured and unstructured data are coming together. The textbook approach assuming a perfect schema and clean data is inadequate. Today's analyses on very large data sets must handle flexible schemas and uncertain data yielding approximate results.

The map/reduce paradigm is gaining acceptance by virtue of its simple scalability. It will likely not be the final answer as its lack of join algorithms beyond sort/merge will prove to be an increasing limitation.

Finally, it was observed that academic computer scientists are not focusing on core database technology any more. Data integration has taken over as the hot topic in the field.

5.2 Impact of Hardware Trends

The number of CPU cores / processing power is increasing rapidly. For this reason alone, databases will have to go massively parallel to consume multi-core CPUs. This parallelization will need to take place at all levels of the software, including both query execution and low-level internal processing.

It is very unclear if and when we might see true optical computers. Such technology, once available, would certainly be very disruptive for databases.

Disks are becoming bigger, and denser. The raw disk transfer rates have improved over the years, but the I/O transaction rate which is limited by disk head movements has stalled; disks are effectively becoming sequential devices. This severely impacts databases, which frequently access small random blocks of data.

Participants emphasized that power and cooling is often overlooked. Databases are typically the single biggest consumer of power in a complete system, primarily due to their spinning disks which generate a lot of heat. It seems inevitable that disk technology will be replaced in the not-too-distant future by a solid-state, no-moving-parts solution. One of the most promising candidates is flash memory. Large-scale deployment of flash, with its random-access abilities, would completely change the way large data sets are managed, having a large effect on how data is partitioned, indexed, replicated and aggregated.

5.3 Conclusions

The next decade will be very interesting. Trends in hardware and software will enable the construction of ever-larger databases, and an increased level of research from the academic community and database vendors could significantly simplify building these systems. These trends include the movement towards massively parallel commodity boxes and solid-state storage and the simultaneous evolution of software towards both lightweight components and specialized analysis engines.

6 NEXT STEPS

There was unanimous agreement that we should not stop with this one workshop, but that this should instead mark the start of a long, useful collaboration. Some agonized over their lack of available time, but the overall sentiment was, "*if you can't spend some time on collaborating, extremely large databases are not your core problem*". Specifically it was agreed that we should:

- conduct another workshop
- try to setup smaller working group(s)
- try to define a standard benchmark focused on data-intensive queries
- set up shared infrastructure, ranging from testbed environments to a wiki for publishing information, including "war stories" of experiences that could be useful to others
- raise awareness by writing a position paper, creating an entry in Wikipedia, and other actions.

6.1 Next Workshop

Participants felt that the next workshop should be similar to the first one. It should be held approximately one year after the first, giving time for progress to be made in smaller working groups and for several projects to accumulate experience, including LHC, PanSTARRS, and the Google/IBM/academic cluster.

The workshop should probably be extended to two or possibly three days to enable more detailed sharing and more extensive discussion. Now that the content and value of the session has been established, attendees will be able to justify additional time away from their offices. At more than one day long, attaching the workshop to an existing conference was thought to extend travel times too much, so it is best kept separate.

"Neutral ground" was thought to be better than a vendor or industry location. While there is undoubtedly selection bias present, the participants felt that holding the workshop in the San Francisco Bay Area minimized travel for the greatest number. We could meet at SLAC again; Asilomar[1] was also mentioned as a possibility.

The number of attendees should not be significantly expanded, as it would be hard to make progress with a much larger group. Participation should remain by invitation only.

The content of the next workshop should focus more on experience sharing to fully bring out commonalities that can be developed into community-wide requirements. If vendors are present, users wanted to be able to question them more.

6.2 Working Groups

The discussion at the workshop was at a relatively high level. It was agreed we need to dive deeper into specific problems. One example that was frequently given was that academics would like to better understand scientific needs. To tackle these problems, smaller dedicated working groups that could meet separately at more frequent intervals would be desirable. A possible agenda for a science/computer science meeting might include:

- developing a common set of requirements for scientific databases including difficult queries, a limited number of desired primitive data types, and a small set of algebraic operators
- developing a mechanism for the scientific community to give academics access to large data sets.

6.3 Benchmarks

Well defined benchmarks have been a good way to describe problems, attract the attention of both database vendors and academia, and drive progress in the field. Existing benchmarks such as TPC-H and TPC-DS are useful, but they do not directly address the usage scenarios of extremely large databases. The group will first try to understand and reach consensus on our common requirements and then define a benchmark that focuses specifically on data-intensive queries.

6.4 Shared Infrastructure

Progress in this area will happen most efficiently if groups can avoid duplication, particularly repeating mistakes. Common shared infrastructure will help here. Initially, we will build a wiki site for groups to publish lessons learned, describe problems, and discuss issues. This site will be moderated but open to all interested parties, including those who did not have a chance to participate in the workshop.

It was also noted that we should try to leverage a recently announced[2] data center, intended for academic use, that has been set up with Google hardware, IBM management software, and Yahoo!-led open source map/reduce software[3].

7 ACKNOWLEDGMENTS

The organizers gratefully acknowledge support from our sponsors: LSST Corporation and Yahoo!, Inc.

8 GLOSSARY

CERN - The European Organization for Nuclear Research
DBMS – Database Management Systems
ETL - Extract, Transform and Load (data preparation)
GFS – Google File System

[1] http://www.visitasilomar.com
[2] http://www.google.com/intl/en/press/pressrel/20071008_ibm_univ.html
[3] Yahoo! also announced a large cluster for use by academia after the workshop: http://biz.yahoo.com/bw/071112/20071112005373.html

HEP – High Energy Physics
IPAC - Infrared Processing and Analysis Center, part of the California Institute of Technology
JHU - The Johns Hopkins University
LHC – Large Hadron Collider
LLNL - Lawrence Livermore National Laboratory
LSST – Large Synoptic Survey Telescope
NCSA - National Center for Supercomputing Applications
OLTP - On-Line Transaction Processing
ORNL - Oak Ridge National Laboratory
PanSTARRS – Panoramic Survey Telescope & Rapid Response System
PNL - Pacific Northwest National Laboratory
RDBMS – Relational Database Management System
SDSC - San Diego Supercomputer Center
SDSS – Sloan Digital Sky Survey
SLAC – Stanford Linear Accelerator Center
VLDB – Very Large Databases
XLDB – Extremely Large Databases

APPENDIX A – AGENDA

Start Time	Duration (mins)	Speaker Moderator	Topic
9:00	20	Jacek Becla	**Welcome**
9:20	40	Dirk Duellmann (LHC) Kian-Tat Lim (LSST)	**Examples of future large scale scientific databases** LHC (20 min), LSST (20 min) The two talks will introduce xldb issues in the context of two scientific communities managing large data sets (High Energy Physics and Astronomy)
10:00	45	Jacek Becla	**Trends, road-blocks, today's solutions, wishes** *Panel discussion, scientific community representatives* Panel will reveal how the scientific community is using and would like to use databases.
10:45	20		coffee break
11:05	85	Kian-Tat Lim	**Trends, road-blocks, today's solutions, wishes** *Panel discussion, industry representatives* Companies will be given 5 min each to give context for their specific xldb problems followed by discussion of how industry is using and would like to use databases.
12:30	60		lunch
1:30	15	Andrew Hanushevsky	**Summary of panel discussions**
1:45	70	Andrew Hanushevsky	**Vendor response** *Panel discussion, vendor representatives* Directed questions from the moderator reflecting previous discussions, as well as open time. No sales talks please.
2:55	20		coffee break
3:15	30		**Thoughts from academia** *Panel discussion, academic representatives* Representatives from academia give their thoughts about preceding discussions, and their vision of how to improve the connection between the research community and practical peta-scale databases.
3:45	60	Richard Mount	**Future**

			Round table discussion, all How to organize xldb-related work most efficiently including leveraging future large scale applications to advance database technology
4:45	15	Jacek Becla	**Conclusions**
5:00			Adjourn

APPENDIX B – PARTICIPANTS

Academia

- DeWitt, David – Univ. of Wisconsin
- Stonebraker, Michael – M.I.T.
- Murthy, Raghotham – Stanford University

Industrial database users

- Baldeschwieler, Eric – Yahoo!
- Brown, Phil – AT&T Labs Research
- Callaghan, Mark – Google
- Das, Aparajeeta – eBay
- Hall, Sandra – AT&T
- McIntire, Michael – eBay
- Muthukrishnan, S – Google
- Priyadarshy, Satyam – AOL
- Ratzesberger, Oliver – eBay
- Saha, Partha – Yahoo! Strategic Data Solutions
- Schneider, Donovan – Yahoo!
- Walker, Rex – eBay

Science

- Abdulla, Ghaleb – Lawrence Livermore National Laboratory
- Becla, Jacek – SLAC
- Borne, Kirk – George Mason University
- Cabrey, David – PNNL
- Cai, Dora – NCSA, University of Illinois at Urbana-Champaign
- Critchlow, Terence – PNNL
- Dubois-Felsmann, Gregory – SLAC
- Duellmann, Dirk – CERN
- Handley, Tom – Infrared Processing and Analysis Center (IPAC)
- Hanushevsky, Andrew – Stanford University/SLAC
- Heasley, Jim – Institute for Astronomy, University of Hawaii
- Kahn, Steven – SLAC/Stanford
- Kantor, Jeffrey – LSST Corporation
- Lim, Kian-Tat – SLAC
- Luitz, Steffen – SLAC
- Matarazzo, Celeste – Lawrence Livermore National Laboratory
- Monkewitz, Serge – IPAC/Caltech
- Mount, Richard – SLAC
- Nandigam, Viswanath – San Diego Supercomputer Center
- Plante, Raymond – NCSA
- Samatova, Nagiza – ORNL / North Carolina State University
- Schalk, T L – U.C. Santa Cruz
- Sweeney, Donald – LLNL, LSST Corp.
- Thakar, Ani – Johns Hopkins University
- Tyson, Tony – UC Davis, LSST Corp.

Database vendors

- Aker, Brian – MySQL
- Bawa, Mayank – Aster Data Systems
- Brobst, Stephen –Teradata
- Ganesh, Amit – Oracle.
- Guzenda, Leon – Objectivity
- Hamilton, James – Microsoft
- Held, Jerry – Vertica Systems & Business Objects
- Hu, Wei – Oracle
- Hwang, JT – Netezza
- Paries, Lee – Teradata
- Tan, CK – Greenplum
- Tate, Stewart – IBM
- Jakobsson, Hakan – Oracle
- Lohman, Guy – IBM Almaden Research Center
- Lonergan, Luke – Greenplum

APPENDIX C – FACTS

Here are some numbers / technologies mentioned at the workshop. Please keep in mind that this is *not* a comprehensive overview - there was no time to delve into details.

Sizes

Currently in production

- Google: tens of petabytes
- Yahoo: tens of petabytes, 100s TB in Oracle, 25TB/day ingest rates
- AOL: few petabytes. 200 TB in PostgreSQL
- eBay: over 1 PB, will have 2-4 in the next 12 months
- AT&T: 1.2 PB. 3.2 trillion rows in single largest table
- BaBar: 2 PB in files (structured data), few TB in database (metadata)
- SDSS: 30 TB

Planned

- CERN (starts 2008): stream of data from detector: 1 PB/sec, most data discarded. Kept: 20 PB/year
- PanSTARRS (starts 2008): few PB
- LSST (starts end of 2014): 55 PB in files (images), 15 PB in database

Database & Database-Like Technologies Used

Currently in production

- Google has large installation of MySQL for ads, combined with reporting. map/reduce plus BigTable plus GFS for searches.
- Yahoo is using Oracle, proprietary column based engine, and open source map/reduce (Hadoop)
- AOL is using mixture of home-grown solutions, Sybase, Oracle, PostgreSQL and MySQL
- eBay is using Oracle where transactions are needed and Teradata for analytics
- AT&T is using home grown RDBMS called *Daytona*
- BaBar used to rely on Objectivity/DB (object oriented database), now uses hybrid solution: structured data in files plus metadata in database (MySQL and Oracle)
- SDSS is using SQL Server

Planned

- CERN will rely on hybrid solution structured data in files plus metadata in database. <2% of data in database (~300 TB, in Oracle).
- PanSTARRS: pixel data in files, everything else in SQL Server. Over 50% of all data in database.
- LSST: pixel data in files, everything else in database-like system, possibly MySQL + map/reduce.