REPORT FROM THE 5th WORKSHOP ON EXTREMELY LARGE DATABASES

Jacek Becla[1*], Daniel Liwei Wang[2], Kian-Tat Lim[3]

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

ABSTRACT

The 5th XLDB workshop brought together scientific and industrial users, developers, and researchers of extremely large data and focused on emerging challenges in the healthcare and genomics communities, spreadsheet-based large scale analysis, and challenges in applying statistics to large scale analysis, including machine learning. Major problems discussed were the lack of scalable applications, the lack of expertise in developing solutions, the lack of respect for or attention to big data problems, data volume growth exceeding Moore's Law, poorly scaling algorithms, and poor data quality and integration. More communication between users, developers, and researchers is sorely needed. A variety of future work to help all three groups was discussed, ranging from collecting challenge problems to connecting with particular industrial or academic sectors.

Keywords: Analytics, Database, Petascale, Exascale, VLDB, XLDB

1 EXECUTIVE SUMMARY

The 5th XLDB (XLDB-2011) workshop focused on emerging challenges in the healthcare and genomics communities, spreadsheet-based large scale analysis, and the challenges of applying statistics and machine learning at large scales.

XLDB-2011 clarified the data-related problems in health-care and genomics. Some problems are general--conceptually-duplicated yet incompatible software, data formats, and usage models. Usage practices are not significantly converging because of a culture resistant to change. Stakeholders maintain a data-scarce mentality in a now data-rich world though some (e.g., those in attendance) have begun to realize the problem. Rapid growth in data scale from new machines and new technologies (DNA sequencing, medical imaging) caught them off-guard but are useful to highlight the lack of scalable tools and the need for stronger, more scalable data management. Unfortunately, funding sources are reluctant to pay for computation.

Spreadsheets in the context of big data were discussed at XLDB, following interest from the past year's workshop. Spreadsheets are individually small, but so popular, numerous, and ubiquitous (esp. in business) that they have become a large problem. Spreadsheets, due to their intuitive interface, are unlikely to be replaced, despite their problems in data quality. They are more like raw data, with no quality-enforcing mechanisms, such as schema, data typing, integrity, and authenticity, and thus are difficult to archive and maintain. The lack of strictness facilitates ease-of-use and reduces friction when exploring and recording ideas. Thus approaches to deal with spreadsheet problems focus on providing spreadsheet interfaces to other technologies better adapted to scale to large data sets, such as Hadoop or parallel RDBMS.

Statistics at large scales are a generally unsolved problem although some specific solutions exist. Statistics software packages do not scale but are used to prototype algorithms before building custom scalable code. Some participants noted that a single software solution balancing usability and scalability is just not practical, but others claimed that enough scaling problems can be addressed to produce an 80% solution. Some common algorithms are difficult to scale due to their computational cost (e.g., supra-linear scaling), so new, cleverer algorithms or more aggressive approximations are needed. Poor communication between statisticians and technologists is a big problem, with statisticians viewing it difficult to find solutions even when they exist, and technologists viewing statisticians

uncooperative in describing their needs and problems. XLDB participants were optimistic about future collaboration and agreed to work towards collecting and curating problem descriptions from statisticians both to help statisticians cooperate among themselves and to help technologists build solutions.

State-of-the-art machine-learning (ML) practice is the extraction of data from their homes in data warehouses, archives, or managed data stores and subsequent feeding of them to specific algorithms. There are three primary approaches for scaling machine-learning. The first is to push logic into databases in order to leverage database optimizations and scalability. Unfortunately, not all logic can be pushed, and the resulting split is inconsistent, messy, and difficult to maintain. Yet databases should be part of the solution; the data cleanup and preparation enforced by databases and their quality controls are important. The second approach is to favor empirical heuristics and avoid sophisticated machine-learning models. This approach argues that existing ML research is sufficient for today's and tomorrow's problems. This last approach echoes the statistics community--prototype at the small scale and building customized code for particular large-scale conditions.

The small, informal atmosphere of XLDB workshops stimulated impromptu discussion of unplanned topics. Interest in free software is growing quickly, but larger organizations balk when commercial support is missing. Service computing architectures are attractive but too expensive when carrier-grade reliability and availability are unnecessary. In discussing communication gaps, we found the gap between SQL and non-SQL enthusiasts is wide and deep, with differences in culture (suits vs. hackers) and in approaches (rigid, well-defined vs. flexible, ad-hoc).

The next steps for XLDB are to reach out to health-care again, to discuss data-integration problems, and to collaborate more with the high-performance computing (HPC) community. XLDB-2012 will be in the San Francisco Bay Area, but another satellite XLDB gathering is likely. Peer-reviewed papers are not being considered for the next XLDB.

2 ABOUT THE WORKSHOP

The Extremely Large Databases (XLDB) workshops provide a forum for topics related to databases of terabyte through petabyte to exabyte scale. The 5th workshop (XLDB-2011) in this series (workshop website: http://www-conf.slac.stanford.edu/xldb2011/Workshop.asp) was held at SLAC in Menlo Park, CA on 20 October 2011. The main goals of the workshop were to:

- reach out to the health care and genomics communities, which were not represented in the past,
- review statistics and machine learning as special topics in big data analytics, and
- discuss spreadsheet-based analysis.

This XLDB workshop followed a 2-day open conference, which was attended by 280 people. This report covers only the workshop portion. Information about the conference sessions, including the presentations, can be found at the conference website (http://www-conf.slac.stanford.edu/xldb2011/ - Jacek Becla, Kian-Tat Lim, and Daniel L. Wang. Facts about XLDB-2011. Technical Note SLAC-TN-12-001, SLAC National Accelerator Laboratory, February 2012).

2.1 Participation

Like its predecessors, XLDB-2011 was invitational in order to keep the group both small enough for interactive discussions and balanced for good representation among communities. XLDB-2011's attendance numbered 56 people representing science and industry database users, academic database researchers, and database vendors. Industrial user representation continues to grow. Further attendance details are on the website.

2.2 Structure

Continuing the XLDB tradition, XLDB-2011 was composed of interactive discussions. It began with panel discussions on new communities: health care and genomics. Next were discussions focused on spreadsheet-based large-scale analysis, followed by discussions on statistics at scale and machine learning. The concluding discussions reviewed the plans for the next XLDB.

3 NEW COMMUNITIES: HEALTH CARE AND GENOMICS

The XLDB-2011 workshop engaged two "new" user communities, genomics and health care, via two representatives from the National Institutes of Health (NIH) and one from GNS Healthcare. Workshop attendees discussed data management and analytics in these communities: the current practice, the biggest problems, the barriers to solutions, and how they and the larger XLDB community could make progress.

Fragmented, small-scale approach to data

Genomics and healthcare communities are very fractured, with no consensus among many groups producing and managing data. Both communities have a pragmatic perspective of computing as a necessary but periphery expense. With little incentive for standardization and unification, data-producing equipment and data analyzing practices vary widely. Commonalities in language, definitions, and practices are scarce, making collaboration difficult. For example, sequencing machines have inconsistent resolutions, file formats, and interfaces that sometimes vary even between releases of a particular machine. The resulting "mess of data" is difficult to work with and divisive to the community. The good news is that people are beginning to notice this "horrible fragmentation."

Some effort is being made to decrease in-house development in favor of more off-the-shelf software (possibly increasing interoperability). The genomics community welcome both not-too-expensive commercial and open source software but find that the commercial systems are "too expensive" and open-source is "not there yet" and "needs time to mature." Thus the community continues to develop their own solutions. In-house solutions are also often developed because requirements and specifications are rarely known ahead of time—by the time they are finally known, a custom, non-elegant, half-baked solution is usually ready for deployment.

The healthcare industry frequently purchases commercial software, such as analytics software, and this results in "huge expenditures," some of which are unfortunately "wasted." Some companies are both *users* and *providers*, such as GNS Healthcare who specialize in building and commercializing custom solutions. Industrial users value commercial support for open source because they can externalize the associated liability to an external company.

Somewhat less fragmentation exists in programming languages. Both communities use Java, R, and various scripting languages. Though not popular, SQL is also an acceptable language. R, a statistics package, is ubiquitous in genomics and used for many popular projects (e.g., the *Bioconductor*, a framework for analysis and comprehension of high-throughput genomics data). R is broadly accepted and appreciated but understood to have poor scalability. The community is accustomed to working around R's limitations, needing more scalable analytical tools but not aware of any better alternatives.

Problems driven by technological advances

The genomics community has an immediate and desperate data explosion problem to solve within 1–1.5 years. The data explosion itself was caused, unsurprisingly, by technological advances. Higher resolutions and higher performance of instruments that are now cheaper by orders-of-magnitude are causing a growth in data production well beyond Moore's law. At the time of the workshop, NIH had collective capacity to produce 1 petabyte per year.

The main problem is cultural and human rather than technological. The biology community has been slow to accept computing as an important part of research. Biologists are not used to accounting for computing and analysis in their budgets. Historically, sequencing has been expensive and its data scarce, meaning that the cost of storing and analyzing data was negligible. Conditions have changed drastically—the National Human Genome Research Institute (NHGRI) reported that it cost $10M to sequence a human genome in 2007 but just under $10K in 2011 (Wetterstrand, KA. *DNA Sequencing Costs: Data from the NHGRI Large-Scale Genome Sequencing Program,* available at: www.genome.gov/sequencingcosts). Hardware infrastructure for biologists has not kept up, especially for most biologists: an estimated half of all grants are forced to use insufficient, non-scalable, fixed compute infrastructure.

Another aspect of the human problem is that both communities lack people capable of grasping the "big data" challenges: one participant estimated that for some 1.4 million "data scientists" in health care, there were only 200 thousand "big data" people. The lack of know-how of existing tools and technologies is a problem as well. A lot of things that were obvious to the XLDB community were completely foreign and unknown to biologists. New

technologies may be needed, but the lack of awareness of and expertise in existing computing is a more immediate problem. Most of the community does not possess the skills to write custom code and integrate off-the-shelf software. Among those with the skills, knowledge is typically shallow and ignorant of deeper system architecture and software construction principles. Sometimes the superficial knowledge is detrimental—one participant quipped "A little knowledge is a dangerous thing."

The biologists who do invest significant effort in computing (e.g., designing data layout in databases) are often dismissed in their community ("structuring data is not science", "writing code is not biology"). Programmers are "second-class citizens." Hospitals seem particularly unappreciative of IT professionals—one attendee told of a sad exodus of talented IT staff from a hospital where they felt particularly unappreciated. One commercial vendor argued that there was "too much democratization" of data analytics tools, claiming that better, commercial options are neglected. The practice of medicine was said to be still largely an "oral tradition" that lacks computational methods.

One new interesting practice not mentioned at past XLDB events was related to the determination of *causality* from data. Health care users particularly need to analyze data causality, that is, which data affect other data and in what ways. They noted that there are too many "known" causalities that are *untrue* in practice.

Future directions

Attendees were optimistic that the cultural problems can be solved. Collaboration should be increased between software engineers and biologists. Bridges are needed between science, computer science, academia, and industry. More partnerships are needed between hospitals and solution providers. One idea mentioned involved implanting engineers in projects as a way of changing the culture. Although one participant was skeptical that biologists would accept this, another cited a Dutch science foundation that required computer science students to work directly on a science problem as a degree requirement. Greater collaboration would reduce investment in the wrong things. For example, IBM invested heavily with a genomics company, but focused too much on the business buyer instead of the scientist, and ended up building models and demonstrating impressive computation while not significantly improving science. Vendor representatives suggested that collaboration could happen on an institutional level so that computing experts do not get pigeon-holed by biologists into system and database administration.

To address the community's lack of expertise, solutions can be delivered as *services* rather than software and hardware that would require (greater) customer integration. In this way, the community can outsource its computing needs to experts and potentially reduce the need for in-house development. It was unclear, however, whether their data analytics can be met in this fashion. Another proposed way to reduce software integration and solution effort is to promulgate a common software stack for science much in the way that web companies have standardized on a LAMP (Linux-Apache-MySQL-Perl/PHP/Python, a popular foundation for application servers) stack. This tactic of raising the degree of shared commonality is used interdepartmentally at Indiana University.

The attendees recognized that XLDB facilitated solutions to a lot of the above problems. The workshop and (more recently) the conference were founded to facilitate communication and advocate development of solutions for large data problems. XLDB has built a human network that can collect use cases, put up a collection of well-chosen cases ("lighthouses"), find common needs across disciplines, match the technical strengths of available solutions with the requirements, produce recommendations, and package them into stacks ("foundation services") usable by multiple disciplines. XLDB has already made a visible impact in technology use in some places (e.g., Exxon Mobil, according to a company representative), and it has the strong potential to influence culture elsewhere.

4 FROM SPREADSHEETS TO LARGE SCALE ANALYSIS

Although individual spreadsheets are not "big data," they contain vast amounts of data in aggregate. The previous XLDB workshop's attendees sought help in managing this data—their widespread usage in storing critical fragments of data could no longer be ignored. This session brought clarity to the problem of spreadsheets in the context of large scale data management and analysis.

While the usage and data volume of spreadsheets is not well known, one attendee estimated that 90% of all business data resides in spreadsheets. Used by practically every computer user, their ubiquity is undisputed. They are often

used for critical multi-billion dollar business decisions, claimed one participant. Some popular uses for spreadsheets include:

- authoritative storage for field data collection,
- tools for computing or presenting summary statistics,
- simple ways of data visualization,
- places for ad-hoc integration of multiple data sources,
- input forms for data entry,
- scratch pads, and
- sandboxes for prototyping analysis techniques.

One participant declared that "the human mind thinks in rows of tables." Hence, the spreadsheet interface model will never be replaced, despite the problems with the way they are used and the way most spreadsheet software operates.

The tabular interface of spreadsheets for editing, visualizing, and manipulating is intuitive and powerful. Powerful functionality is usually included by default and more specialized functionality, such as text analytics or statistical processing with R, is well-integrated. Registered Microsoft Excel users were estimated to number 500 million (most of whom are not data professionals), and unregistered users equally as many. Some data sets, such as the Statistical Abstract published by the U.S. Census Bureau, are published in spreadsheet form. The simple row and column structure impose minimal constraints and do not interfere as users input, edit, manipulate, and interact with their data. Spreadsheets make great playgrounds for exploring data, developing algorithms, and constructing models. (Algorithms developed by analysts inside spreadsheets are often completely rewritten by programmers and converted into "real", repeatable pipelines to enable execution on large data sets. Large data sets are not stored in spreadsheets since their unified input/edit/visualize table interface is unwieldy beyond some number of rows and columns.)

Unfortunately, when the interface is loose, so is the data. Data types, units of measurement, and other semantic meanings are not directly stored, so data are re-interpreted according to user specifications for each new formula or chart. Spreadsheets do not specify or impose schema, as traditional databases do. Spreadsheet data are copied frequently, sometimes with transformations or tweaks, sometimes as a shim for integrating with other software, and sometimes for sharing. With proliferation of copies the authoritative, canonical version is often unclear. These problems of schema, data types, integrity, and authenticity are inherent to the spreadsheet model of computation. Capabilities difficult for databases, such as data provenance, security, and reproducibility, are doubly difficult with spreadsheets. All of the above make stewardship of data stored in spreadsheets very expensive or impossible.

Participants agreed that solving these problems should preserve elements of the spreadsheet interface but that the computation and storage should be moved off the desktop. One approach is to integrate data from existing databases, data warehouses, and Hadoop clusters into a spreadsheet interface (the Datameer Analytics Solution is one such example). Another approach is implemented as a scalable cloud-based spreadsheet system (Google Fusion Tables (GFT) is an example of a cloud-based spreadsheet, allowing storage of tabular data for search, visualization, and collaboration). No approach seemed dominant, and more implementations are forthcoming. Neither of these approaches addresses the problem of data in existing spreadsheets, and while no solution is widely available, a participant from the University of Michigan demonstrated a sort of spreadsheet search engine, which is able to infer semantic meaning and structure from a large spreadsheet collection and answer free-text queries using the resulting index.

5 STATISTICS AT SCALE

The workshop participants agreed that the general problem of performing statistical analysis at large scale remains unsolved, despite reported solutions for specific sectors from SAS and other commercial vendors. Just as other computing applications have struggled to cope with growing data intensity, statistical analysis software packages like MATLAB, R, and SAS are found to be ineffective or inapplicable at large scales. The paradigm suggested was that a statistical methodology should be developed using these tools on a data sample prior to re-implementing the solution on a more scalable platform like Hadoop, much as software is often prototyped using a slower but more dynamic programming language and then reimplemented using a faster, "production" language. Echoing this practice, John Chambers, creator of the S statistical programming language, pointed out that statistics packages were

designed to allow statisticians to focus on the problem rather than details like scale and efficiency, which would be solved in a reimplementation anyway. Thus statistics software packages, like spreadsheet packages, should be used as a prototyping playground, with the heavy lifting to be done elsewhere. This also resonated well with a conclusion from previous XLDB workshops that "no single software system is a complete solution."

Unfortunately, large-scale data-intensive computing is inherently non-trivial, and participants sought ways to integrate more scalable computing platforms with statistics software. SAS's user-defined function (UDF) capability allows delegating functionality to a variety of database backends, including scalable parallel databases, such as Teradata. Similar computation-outsourcing plugins are available for R (noted in particular was the Rhadoop project [including rhbase, rhdfs, and rmr] of Revolution Analytics, a company founded to provide commercial support and development of R. The company was also developing a parallel implementation of R), but participants quickly noted that those are still young and need much work. Pushing processing to an external scalable backend is somewhat successful, but several claimed that making more and more computing resources available through statistics software, even if achieved, would not be a full solution.

A larger problem, claimed some, was the computation of algorithms with supra-linear time cost, i.e., those whose cost scaled with the square or cube of the data size or even greater. While more computational resources would speed their execution, their overall computational cost would remain unworkable. More work is needed to develop more efficient methods, either by novel implementation techniques such as Strassen's algorithm for matrix multiplication or by using stochastic methods to reduce the computational data size. In praise of good approximations, a participant noted that the web search problem was essentially an eigenvalue problem and that Google's PageRank provides efficient approximation of the largest eigenvalues.

Generally, the largest barrier to solving the scalable statistics problem is poor dissemination of knowledge of existing approaches. Statisticians were unaware of solutions that may already exist—R's software library is immense but often bewildering to most of its users. Software developers lacked enough details of statisticians' problems, making it difficult to identify and develop useful solutions. Some computing researchers complained that those with the problems are often reluctant or unable to release details due to security concerns (especially in medicine due to patient privacy) or competitive concerns (of both for-profit companies and academic researchers). A Microsoft representative noted that scientific communities have not made it clear what features are missing from existing tools. Some proposed the collection and curation of representative problems and case studies with sufficient detail that (a) statisticians can find problems (and accompanying solutions) similar to their own and (b) developers can evaluate their ideas and prototypes against concrete specifications. Finally, a benchmark or formal statement of a statistical "grand challenge" would spur new research, as the *PennySort benchmark* did (see: *Performance / Price Sort and PennySort*, by Jim Gray at al., MS-TR-98-45), push existing systems into addressing the yet-unsolved problems, and expose how rapidly vendors' technologies can solve the statistics-at-scale challenges.

6 MACHINE LEARNING

XLDB-2011 investigated the problems and considerations of machine learning (ML) in the context of large data volumes.

One interesting idea was that ML processing can take place entirely within databases, instead of extracting data from the database into an external statistics package. If ML primitives are to be implemented in the database layer, processing can leverage common database optimizations such as parallelism, caching, and improved I/O scheduling. The implementation of a data mining model in a relational database was described, mapping model specification to model-table creation, training to table loading, learning to querying, and prediction to a join operation between an input table and the model table. Yet the implementation showed that databases are poorly matched for performing all phases of ML processing, even though they can be used to accelerate significant portions. Participants agreed ML should exploit data within databases and thus leverage the considerable cleanup, normalization, and other preparation necessary to load data into strict schemas.

Another participant believed that the current data-abundant world is well-suited to empirically-trained models rather than sophisticated predictive algorithms and maintained that well-tuned, well-optimized heuristics are more effective in practical settings. LinkedIn's "People you may know" feature was cited as such an example of a collection of heuristics well-executed at scale. There was a sentiment that academic research in machine learning is already

adequate to apply to general real-world problems, but real usage is uncommon and scattered—though there is evidence of successful ML usage in several places. One participant noted that heuristics are not grounded and their results cannot be used in diverse situations, but others felt they are usually adequate, citing extensive use in medicine.

Representing ML models was cited as a challenge. The Predictive Model Markup Language (PMML) is one of a few specifications, but no standards have been widely-adopted. Quality control was cited as another problem—one participant wished for quality measures for each step of processing to aid in understanding the result.

Participants discussed more general considerations of deriving answers from data. The multi-hypothesis pitfall (given sufficient data, any arbitrary hypothesis can be supported) is a real danger in data-abundant environments. Another problem is that algorithms often operate on data sets assuming they represent the complete, closed world—a dangerous assumption because empirical data are usually more accurately considered a partial sampling.

The practice of prototyping in one language or software environment and reimplementing in another is reiterated in the context of machine learning. Reimplementation is not only an opportunity to improve the algorithm but also an opportunity to introduce errors or misinterpretations. Algorithm creators and reimplementors are usually different groups of different specializations (i.e., statisticians and computer scientists) who communicate poorly due to differences in jargon, perspectives, and priorities. Most were not optimistic that prototype-and-reimplement can be eliminated in favor of a unified development process but suggested addressing the barriers between the different groups. Interdisciplinary groups were suggested over collaborating groups. A collection of data with use cases was again suggested as something that would steer computer scientists towards building better, more suitable tools while providing reference “best practices” for data scientists and other domain experts.

The XLDB community could help with enabling scalable ML by identifying and connecting with people who can represent their community with an understanding of both their domain and its computing. Also helpful would be the identification of a prototypical case, such as bi-clustering on the EXPO oncology data set (few GB) or analysis of the 200TB sequence data at google.org.

7 OTHER TOPICS

XLDB workshops foster intense discussions that often diverge toward unanticipated topics, and a sampling of these topics is provided below.

Free (libre) software was discussed as a solution tactic and as a development model. Representatives from larger industrial entities noted that commercial solutions have been strongly preferred historically, but that acceptance of free software is growing and not as frequently dismissed as immature. For the larger entities in non-computing industries, free software is only viable when accompanied by a commercial entity that provides support and absorbs liability.

Service computing is a promising method to satisfy storage and computation scalability needs, with some reservations. While current pricing is reasonable for commercial for-profit entities, another alternative is needed for academic usage where low cost is prioritized over high reliability and performance. Amazon Web Services is reportedly investigating such an alternative.

In discussing analytical models, participants recognized that a database computation model, where a question is posed in terms of a declarative query to an engine executing close to stored data, is effective for a large fraction of analytical processing. However, they also pointed out that iterative processing techniques, such as those that compute a result after some (possibly non-deterministic) number of repeated steps, are poorly supported by databases and require their own custom implementations.

Probabilistic answers to queries, that is, presenting a single answer as a probability mass function or probability density function, were reportedly desired by statisticians to guard against misinterpretation of results. However, no off-the-shelf solutions were mentioned, and one participant cautioned that implementing such an engine was much more difficult than one would expect. One approach was described in a poster at the XLDB conference: a prototype implementation that computed a probabilistic answer by duplicating database instances for each possible outcome.

The gap between communities using SQL and those not using seemed nearly unbridgeable to most participants. Hatred of SQL stems from the accompanying baggage accompanying most SQL-speaking database software, e.g., transactions and schema, but other problems are rigidity and inflexibility, e.g., the lack of support for fancy hacks in SQL, and the difficulty of embedding within C or other programming languages. Participants felt that while conversion and reconciliation are not likely, the communities can learn a lot from increased communication and knowledge exchange.

The use of data in high-performance computing simulations was split into two large categories. The first is concerned with monitoring large simulations' produced data streams for early error detection. The second is the "offline" analysis after such simulations complete and frequently not considered for execution on "big-iron" supercomputers.

8 NEXT STEPS

As in the past, a small portion of the workshop was devoted to future planning.

The next XLDB event should again reach out to the healthcare community. There is much more to explore and discover about large data problems in health care, whereas genomics and web-scale problems were well-covered this year. More general diversity was requested for the next event. Industrial communities suggested included mobile telecommunications, manufacturing, in-flight data (e.g., Boeing), and national intelligence (e.g., DARPA). Hardware vendors, such as Intel, were requested so they could offer their long-term plans and perspective on emerging trends to the attending large data communities. Input from the venture capitalist perspective, perhaps via a panel discussion, was also requested.

The most-demanded topic was data integration. The data integration problem is one of the biggest unresolved challenges with few who understand the problem and even fewer who are working on solutions. Another topic was cloud computing in terms of costs and tradeoffs for data-intensive (not typical high-performance computing) usage. Other topics were database support for per-query schema (in response to schema-less computing in Hadoop) and array databases.

By far, *the most highly-demanded* new future activities for the XLDB community were:

1. collecting test cases (data and corresponding application software),
2. collecting use cases/challenges,
3. setting up a single repository to document/publish collected information (e.g. test cases and use cases from above), and
4. specifying “reference architectures” that detail particular hardware/software combinations for data-intensive solutions.

Attendees were interested in working more closely with the HPC community: many believed there are numerous lessons that the HPC and the XLDB communities could learn from each other. The best idea was for XLDB representatives to attend an HPC gathering such as one of the Supercomputing conferences. XLDB itself is too small to accommodate a large HPC contingent, and attendees felt that a small HPC contingent might feel uncomfortable or alienated.

The next conference should remain substantially similar to XLDB-2011 in overall length, balance between topic diversity and focus, and organization. Possible improvements suggested including an additional specialized workshop day and demonstrations (possibly during the pre-dinner reception). The community felt strongly that introducing peer-reviewed papers was *not* a good idea since they could dramatically reduce the presentation quality and suppress the open, uncensored dialog. Satellite workshops, such as XLDB-Europe in Edinburgh, were considered beneficial, but organization of a workshop located remotely (e.g., in Asia) could be difficult due to distance, language barrier, and visa issues (especially if organized in China). Given the limited financial and human resources of the XLDB core team, the tasks will need to be carefully chosen and balanced to maximize the benefits to future XLDB events and activities.

9 ACKNOWLEDGEMENTS

The XLDB-2011 workshop was sponsored by eBay, Vertica/HP, Chevron, SciDB&Paradigm4, MonetDB and IBM.

The XLDB-2011 was organized by a committee consisting of Anastasia Ailamaki (École Polytechnique Fédérale de Lausanne), Jacek Becla (SLAC, chair), Peter Breunig (Chevron), Bill Howe (University of Washington), David Konerding (Google), Samuel Madden (MIT), Jeff Rothchild (Facebook), and Daniel L. Wang (SLAC).

This work was supported by the U.S. Department of Energy under contract number DE-AC02-76SF00515.

10 GLOSSARY

ETL – extract-transform-load
DARPA – Defense Advanced Research Projects Agency
DOE – Department of Energy
GFT – Google Fusion Tables
GNU – GNU's Not Unix
GPL – GNU General Public License
HPC – High Performance Computing
LAMP – Linux-Apache-MySQL-Perl/PHP/Python
ML – machine learning
NHGRI – National Human Genome Research Institute
NIH – National Institutes of Health
PMML – Predictive Model Markup Language
RDBMS – Relational Data Base Management System
UDF – User Defined Function
XLDB – eXtremely Large Data Base

(Article history: Received 3 March 2012, Accepted 13 March 2012, Available online 18 March 2012)