REPORT FROM THE 3rd WORKSHOP ON EXTREMELY LARGE DATABASES

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

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

ABSTRACT

Academic and industrial users are increasingly facing the challenge of petabytes of data, but managing and analyzing such large data sets still remains a daunting task. Both the database and the map/reduce communities worldwide are working on addressing these issues. The 3rd Extremely Large Databases workshop was organized to examine the needs of scientific communities beginning to face these issues, to reach out to European communities working on extremely large scale data challenges, and to brainstorm possible solutions. The science benchmark that emerged from the 2nd 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, VLDB, XLDB

1 EXECUTIVE SUMMARY

The 3rd XLDB workshop (XLDB3) focused on reaching out to communities underrepresented in the first two workshops, improving the science benchmark, and reviewing available large-scale solutions. The new communities brought to XLDB3 expressed strong needs for petascale solutions: radio astronomy's SKA survey is preparing to deal with a data set that might exceed an exabyte; geoscience needs solutions for integration and unification of their diverse data sets; and biology faces a data explosion in microscopic imaging and protein and genomic sequencing. LHC's largest headaches reside in managing the heavily-accessed metadata for their petascale data. Nokia reported that industrial users urgently need a robust petascale analytics platform, showing that their efforts (and funding) for analytics are strikingly large even when compared to efforts in large scientific projects.

To achieve extremely large scales, partitioning/chunking and distribution of data is required. Data distribution could be hierarchical, as in HEP's hub-and-spoke model, or non-hierarchical, as in geoscience and others. A significant fraction of scientific data is image-based and can be naturally represented as n-dimensional arrays. These data sets fit poorly into relational databases, which lack efficient support for the concepts of physical proximity and order, and so they are typically stored in array-friendly formats such as HDF5, netCDF, FITS, or casacore. Scientific data sets often need to be integrated from a large variety of sources, transformed, regridded, aligned, and calibrated before they can be analyzed. Performing QA and maintaining data consistency pose new challenges at the petascale level and often require using advanced techniques such as machine learning. Long term data preservation is hard but necessary and is being seriously investigated. No comprehensive, petascale, freely available database solutions exist yet, so most large-scale users continue to write custom software. User-defined function (UDF) interfaces are helpful but provide only a minimal level of software reuse. Scientific process flows are highly optimized for maximum throughput and minimum cost, often at the expense of flexibility and response time. Both industrial and science users observed that the majority of their resources are consumed by a small number of sophisticated users. Despite the huge potential of mining raw data, such as transaction logs in industry or raw pixel data in science, all participants reported throwing away most of this valuable data, as it is too expensive to keep. Centralizing analysis requires a paradigm shift within the science community, as the desire for owning and controlling data by individual scientists stands in the way. Funding for scientific data management (SDM) is often misallocated: examples include overfunding data *collection* at the expense of underfunding *analysis* or cutting SDM funds in favor of funding science directly. Finally, SDM appears to evolve at a much slower pace than its industrial peers, due to tight funds, legacy software, and inertia within large communities centered around big projects.

The map/reduce (MR) model, which has become popular in industry, was discussed. The ease of expressing queries through a procedural language and the availability of a free open-source system (Hadoop) were believed to be among the strongest points of MR. Frequent checkpointing of MR limits performance but is critical for handling failures that can wreak havoc with RDBMSes' optimistic assumptions. Strict enforcement of data structures in RDBMSes has led users with poorly structured and highly complex data to avoid databases. Luckily, the RDBMS and MR communities are quickly learning from each other; each community is fixing its deficiencies and adding missing features. In practice, they appear to be rapidly converging.

Several solution providers presented their thoughts on terascale and petascale analytics. MonetDB presented a successful port of the SDSS multi-terabyte database. Cloudera discussed activities to support the Hadoop community. Teradata explained new techniques involving migration of data to appropriate (faster or slower) media based on frequency of access. Greenplum discussed dynamic re-mapping of a pool of servers to warehouses. Astro-WISE presented their system. SciDB demonstrated a from-scratch prototype supporting an n-dimensional array data model, running in a shared-nothing environment.

A first draft of the science benchmark concept, introduced at the previous XLDB workshop, was discussed. The draft covers raw data processing and derived data analytics in the context of an array data model. The next steps include adding extra scaffolding, broadening the team, and expanding the scope to cover additional data models.

It was agreed that the next XLDB workshop, XLDB4, will be held in the fall of 2010 in Silicon Valley in the United States. It will attempt to reach out to remaining underrepresented communities, and industry presence will be increased. Biology, geoscience, and HEP will provide their use cases shortly after the XLDB3 workshop. Applying for funding from the European Commission for Europe-based XLDB and/or SciDB activities through "FP7" proposals will be considered.

2 ABOUT THE WORKSHOP

The Extremely Large Databases Workshops (XLDB) provide a forum for topics related to databases of terabyte through petabyte to exabyte scale. The 3rd workshop[1] in this series was held in Lyon, France, from August 28 to 29, 2009. The main goals were to:

- exchange information with communities outside the USA,
- invite and involve more science disciplines, especially those underrepresented in the past,
- review existing XLDB tools and solutions and discuss how to move the state of the art forward.

2.1 Participation

Like its predecessors, XLDB3 was invitation-only. This kept the attendance small enough to enable interactive discussions without microphones and ensured an appropriate balance of participants among the communities. Fifty-two people attended, including science and industry database users, academic database researchers, and database vendors. Compared to past XLDB workshops, a smaller fraction were industrial users, owing to difficulties in finding the appropriate people from European companies. Further attendance details are on the website.

2.2 Structure

Like past workshops, XLDB3 was chiefly composed of highly interactive discussions. These included sessions on complex analytics, scalable architectures, and various XLDB solutions. Particular focus was given to issues for scientific communities underrepresented in the past, including geoscience, radio astronomy, and biology, as well as high-energy physics (HEP), which was well-represented due to CERN's proximity.

[1] Website: http://www-conf.slac.stanford.edu/xldb09

3 DATA USER COMMUNITY PERSPECTIVE

XLDB3 involved several new user communities: geoscience, biology, radio astronomy, and a representative from the telecommunications equipment and service industry (Nokia). With the geographic proximity to CERN, it was natural to delve deeper into the needs of HEP, particularly the Large Hadron Collider (LHC) experiments. Although overtures were made, the workshop will need to continue to work to draw attendance from more communities such as chemistry and the oil and gas industry.

3.1 Specific user communities

Many science and industrial users already, or will soon, manage petabytes of data, some in a database but most outside. HEP reached the petascale with projects like BaBar, D0, and CDF. Optical astronomy will reach this data size as surveys such as PanSTARRS and LSST come on-line. More science domains are approaching the petabyte scale, and a few are even planning for exabytes of data. Approaches and solutions for smaller data scales, which are strained at current data volumes, will soon be non-functional and must be redesigned.

3.1.1 Radio astronomy

Radio astronomy's Square Kilometer Array (SKA) survey, which has a planned construction start date of 2015, will collect data from over 3,000 dishes plus other receptor technologies covering an area of one square kilometer at a rate of`100s of terabytes per second. SKA's current major challenges are to estimate costs for sponsors, to determine the feasibility of potential requirements, and to find ways to reduce the data volume. With the current level of reduction, SKA estimates its persisted data will exceed an exabyte over its 20+ year lifetime. The community is exploring ways to automate analysis, as the current manually-driven methods will not cope with the new data volumes.

3.1.2 Geoscience

Unlike other data intensive communities, geoscience has no “very large” projects[2]. Instead, it is a diverse “system of systems” consisting of many disciplines such as solar/heliospheric physics, planetary sciences, geology, atmospheric and ocean sciences, tectonophysics, seismology, and hydrology. Data sets from individual observatories (surface, orbital, etc.) or instruments may be multi-terabyte-sized. An important challenge is the integration and unification of these diverse data sets to facilitate their access and analysis in larger, more coherent units.

3.1.3 Biology

Biological data sets, at a high level, are organized like those in geoscience – they consist of a large number of relatively small, independent data sets. However, two notable exceptions exist: (a) microscopic and other forms of imaging and (b) protein and genomic sequencing. Both are now approaching the petascale. For example, 20 samples of 20,000 genomes, each containing 3,000,000 bases, amount to 1.2 quadrillion floating point numbers. The growth in sequence data is fueled by rapid decreases (as much as a factor of five per year) in sequencing costs, and the resulting rapid expansion in sequencing machines from a few per city to a few per hospital. Controlling access to data in a way that complies with ethical and privacy regulations is a big problem.

3.1.4 HEP/LHC

LHC has now started data taking[3], and will soon generate some 15 petabytes of data per year, with a matching amount of data from simulations. The vast majority of this data is composed of events stored in files, with tens of terabytes of database storage used for auxiliary information, including calibrations, detector and accelerator conditions and configurations, metadata for event selection, provenance, catalogs, and file and job management metadata. The main database challenge for the LHC experiment teams lies in managing the metadata needed to select, understand, and process the event data, including substantial amounts of information that vary with the time of each event. While such data amounts to only tens of terabytes, ensuring scalable access to it for a large number of simultaneous, distributed jobs poses non-trivial challenges. Selections based upon event-level

[2] Such as LHC in HEP, LSST in optical astronomy, and SKA in radio astronomy.

[3] It started shortly after the workshop, before this report was written.

metadata are typically time-varying and combinatorial and so also tax the capabilities of traditional relational databases even at terabyte scales.

3.1.5 Nokia

Nokia was the only industrial user officially presenting at XLDB3. Business analytics at Nokia are typically performed using a combination of Teradata and Oracle, but the company is simultaneously trying many other approaches, including normalized and denormalized databases, in-memory databases, Hadoop, and home-grown software. The amount of effort (and funding) going towards analytics is strikingly large compared to efforts in large scientific projects. This effort is driven by the huge business value of having agile, flexible analytics. The system has to be capable of adjusting to rapidly changing needs; new data marts are built daily; and new sources of data are integrated weekly.

The company is already dealing with petabytes of data, and it expects to reach the level of a few tens of petabytes as soon as next year. The system is managed by a team of nine highly skilled US-based engineers, augmented by some 200 off-shore developers in Asia.

3.2 Data distribution architecture

3.2.1 Distribution models

Data can be distributed in three major models: (1) centralized, non-distributed, (2) hub and spoke, hierarchical, and (3) non-hierarchical, fully-distributed. The centralized model, where data is kept in a single place (data center), is technically the simplest for access and for backup. Data loss can be avoided by single site backups and offline storage. Funding concerns affect data distribution, as funding has been easiest to obtain for locally-controlled data repositories rather than centralized ones. In the hub-and-spoke model, there is a central warehouse with a master copy of less-processed data and spokes (data marts), which extract portions from the master and apply additional processing. The hub-and-spoke model is advantageous in politics/funding but can be problematic when access to products across multiple sites is needed or when data are updated. The fully distributed model, where data is produced independently at many sites, may be the most funding-friendly, but the lack of organization or standards (even de-facto standards) make data integration a serious problem. This model, prevalent in geoscience and much of biology, happens naturally when no major experiments or collaborations dominate. Industrial users reported use of all three models.

3.2.2 Problems

To manage large data sets, most data-intensive users “chunk” data into manageable pieces and distribute them across multiple data centers. Determining the right distribution scheme is non-trivial, as there are many important implications that need to be considered, ranging from data locality, latency and performance, through cost, to data recoverability. Excessive centralization leads to insufficient protection against failures, while excessive distribution is counter-productive: it leads to redundant data storage and expensive WAN transfers. In some cases, especially in industry, data distribution must be abstracted from end-users; e.g., a system must gracefully recover even from a failure of an entire data center. In science, this level of transparency is not usually required.

3.2.3 Practice

Industrial users tend to build geographically distributed data centers. In typical practice, data is replicated in at least two locations, and in case of a failure of a data center, another data center (or several) take over the traffic. The hub-and-spoke topology is frequently used, consisting of a few large centers and many smaller data marts.

Different science domains have adopted different models. Most common is a tiered approach, a form of hub-and-spoke pioneered by the HEP community. Data is organized into hierarchical data centers: a single tier 0 center (e.g., CERN), large analysis centers at tier 1, medium and small data centers at tier 2 and tier 3, and small tier 4 sites (which may be as small as a team of a professor and a few students). Each tier supports different roles and different forms of access. The tiered model is being considered by both optical and radio astronomy and seems to function especially well for large collaborations with distributed funding.

The distributed model is seen in the geoscience community, with a large number of different, largely independent sites. Services such as OPeNDAP[4] are built on top of these sites that aggregate, organize, and virtualize access, much as search engines organize the web. Such distribution has happened naturally due to highly distributed funding practices. The resulting diversity of data formats and access methods is a serious problem hindering wide data use. About 50 years ago there were attempts to introduce large data aggregation centers in geoscience, with only partial success: these centers became just archive centers instead of data access centers as originally planned. Though there are some new attempts to restart this effort now, the distributed nature of funding and the research itself make a unified effort hard.

Within the larger biology community, the genomics community requires that all raw data used to publish must be archived and publicly available, except where personally identifiable information is involved. This led to the establishment of three large data centers, each archiving and providing data access. This model seems to work well. One of the main "headaches" reported by these centers is a widely varying level of expertise on the receiving end: the end users range from a small number of experienced scientists to millions of untrained clinicians.

3.3 Data formats and models

A significant portion of scientific data is image-based and can be represented as n-dimensional arrays. This is true for almost all of the geoscientific data, most astronomical (both optical and radio) data, most biological data (especially the microscopic images that are experiencing the most growth), and medical images. In a large proportion of cases the images have spatial dimensionality (x/y, longitude/latitude, and right ascension/declination) and time dimensionality (a time series). Of the domains represented, only HEP and parts of biology reported significant volumes of non-image data. HEP deals with uncorrelated events, where an event is a complex structure including tracks of particles, and the non-image parts of biology deal with big graphs and sequences.

All sciences that deal with images store them as flat files, using popular formats such as HDF5, netCDF (geoscience, biology), FITS, and casacore (astronomy). These communities had considered or experimented with relational database management system (RDBMS) solutions for images but ended up reverting to flat files and using the RDBMS only for metadata. The two main difficulties with RDBMSes are:

a) images fit poorly in a relational data model: their spatial and temporal dimensionality and the proximity and ordering of their pixels are essential and not easily represented in a set-oriented relational model;
b) naive image data chunking hampers parallelization: many operations (e.g., near neighbor search or regridding) require looking at adjacent pixels, which often are contained in an adjacent chunk that may be stored on a different node.

RDBMSes fit better for derived data distilled from images. Even there, however, the attendees noted that SQL is nice when the data are known to exist and to be computable, but that scientific discovery often operates where those are unknown.

The communities noted that data formats are usually driven by data producers (due to funding arrangements), causing data to be archived optimally for data storage but poorly for retrieval and analysis. Data is often stored without sufficient metadata, which makes it impossible to interpret the data. The typical non-uniformity of catalogs and inconsistent level of services make it difficult to find data. Finally, domain-specific data structures and formats further complicate data use.

3.4 Data integration

Data sets often need integration before they can be used. All represented communities noted this need in some capacity, with geoscience having the most sophisticated need.

[4] http://opendap.org

Geoscientific data can come from a large variety of sources: ground observatories, mobile stations, sensor networks, aerial observers, simulation models, etc. The set of data for a given location may have different resolutions, different sample rates, different perspectives, or different coordinate systems and therefore must be transformed, regridded, aligned, and otherwise unified before they can be analyzed. Current tools provide virtualization and aggregation services, but they are spotty and insufficient. As a result, sophisticated integration is generally rare, though the resultant value is tremendous.

3.5 Data calibration and metadata

One of the problems brought up by all scientific users was the issue of data calibration. The calibration parameters of an instrument are essential to correctly interpret its data. Calibration data usually fits the relational data model and is often heavily indexed to enable complex analysis; thus managing it is a database issue.

In HEP, the data set containing calibration and configuration information is the hardest to efficiently organize and provide efficient access to, despite the fact that it is less than 0.05% of the bulk data set.

The geoscience community strongly argued that their calibration data may be unreliable and evolve over time, forcing the community to preserve all raw data, some of which could have been discarded otherwise. An example was the initially incorrect image placement of hurricane Katrina which occurred due to coordinate misalignment.

In biology, the main problem is less related to “calibration” and more to lack of sufficient languages and tools that unambiguously describe metadata. There can be ambiguity even when using the same words to describe actions, areas, and states, such as “which part of an animal was cut out” or “what stage of a disease a given piece exhibits.” These natural-language descriptions also become cumbersome in large data volumes.

By contrast, in astronomy, dedicated calibration data are rarer. Instead, astronomical data are self-calibrated, where the data itself are used to do calibration. That is only possible because the sky is to a large extent “empty.”

Scientific users also pointed out the ever-changing nature of metadata, noting that good research often leads to new metadata.

3.6 Data consistency and quality

Scientific and industrial users noticed new challenges in assessing data quality and maintaining data consistency at the petascale. Referential integrity was insufficient – data from multiple sources may be individually good but collectively inconsistent. While some users have attempted analysis of non-quality-checked data, the “slightly garbaged” results were unacceptable. Users also reported that postponing quality assurance in system development was always a bad idea and significantly increased cost: Nokia reported that not treating QA sufficiently seriously resulted in a cost increase of 40% plus delays of millions of dollars per month.

There is a tendency to abandon strict consistency checking and ACID enforcement in the petascale regime. The biology representatives reported that data often have no built-in methods (not even checksums or hashes) to verify integrity, and petascale data means that disk writes with extremely small error rates (e.g. 10^{-15}) can still be untrustworthy.

Visual image inspection cannot be used for petabytes of data, so scientists must develop new solutions to fully automate data QA. Such solutions often require sophisticated techniques such as machine learning.

3.7 Data preservation

Data preservation is an important topic. Old data are needed to determine baselines and understand long term variability. In some cases, scientists have tried to go back as far as data stored on VHS tapes since they may contain scientifically important information. Industry users noted that old data are often useful for forensic analyses.

Often, the real value of a given data set is not known until much later. Insufficient funding often prevents analyzing all collected data – only about 0.5% of all collected geoscience data has been examined because the community does not have funding and appropriate tools to examine the rest.

Numerous efforts to solve the problem of long term data preservation are underway, such as various study groups (e.g. see http://www.dphep.org/), focused workshops, and dedicated funding for work towards sustainable digital data preservation (e.g. the NSF's DataNet).

Adequate standards are one of the key elements of successful data preservation. There are often too many data standards, as evidenced by the joke, "the good thing about standards is there are so many." Sometimes interpretation of otherwise pristine data is impossible due to lack of metadata – this is the case for some data gathered about the moon, for example. Therefore, well-defined, relatively static schemas such as those inside databases or in structured files such as HDF5 or FITS are important for data preservation. Long-term preservation becomes more difficult in systems such as map/reduce that may lack official schemas and have data format logic scattered throughout the analysis code.

The rapid obsolescence of electronic formats and media is of significant concern. Some users reported falling back on paper-based archives – dumping HDF metadata in ASCII and printing it on paper whose lifetime is empirically known and longer than the design life of many electronic media.

Services such as Amazon's Simple Storage Service[5] could be used to preserve scientific data, but funding agencies often prefer to keep data on servers they have more control over.

3.8 Custom software

Most invitations for XLDB3 were to groups underrepresented in the previous workshops, and the resulting overlap in attendance with XLDB1 or XLDB2 was small. Yet the attendees repeated a message that had been loudly voiced in the past: everybody is writing custom software and reusing little. This is primarily because there are no comprehensive petascale solutions freely available to use or reuse. Custom software built by petascale users includes entire systems like ROOT[6], glue software (converters, services), storage resource managers, pipelines and workflows, provenance trackers, specialized indexes, and add-on features like progress indication/estimation and query suspension.

Some amount of reuse is facilitated by User-Defined Function (UDF) APIs in databases, which allow custom analytics to bypass SQL interfaces and be inserted into servers where they can operate close to data. UDFs can be expressed in more familiar procedural languages, which simplifies complex queries. The Sloan Digital Sky Survey (SDSS) depends heavily on UDFs in nearly all of its queries[7].

Porting code to UDF interfaces may not be easy, however. Algorithms are often already written to different interfaces and packaged into well-tested and documented, community-approved libraries for use with tools such as IRAF[8] or MATLAB. Frequently these tools and libraries are run "next to" a database cluster, that is, they are run on machines with high bandwidth connections to the DBMS. A similar approach considered by some industrial users is to run internal software "next to" a map/reduce cluster rather than migrate everything into map/reduce.

It is very unlikely that the collective XLDB community has simply overlooked available tools. The Nokia representative explained, "There is not an analytics tool that we would not try. RDBMS normalized, RDBMS

[5] http://aws.amazon.com/s3/

[6] http://root.cern.ch

[7] for a list of SDSS UDFs, see *functions* and *procedures* at http://cas.sdss.org/dr7/en/help/browser/browser.asp

[8] http://iraf.noao.edu/

denormalized, Hadoop, statistical tools – we tried them all.” While industries can afford to “let a thousand flowers bloom,” scientific users have more limited budgets and usually decide to build custom solutions rather than risk running into a tool's limits without the ability to fix them cheaply. The consensus however is that “managing data wastes scientists' time and money,” and off-the-shelf solutions are much preferred.

To overcome the re-invention problem, collaboration between key people from different domains is essential. Building such collaborations with appropriate funding may be more difficult than the technical challenge, though.

3.9 Large-scale process flow

Industrial users have built systems to perform *ad hoc* analytics even at extremely large scales, but these systems are generally too expensive for academics. Scientists instead have traditionally built systems that are optimized for throughput, using careful planning and coordination to handle the most intensive and bulky analyses with limited hardware and simple software.

In HEP, the undisputed leader in analytics scale, scientists are organized into strict groups. Each group vets all publications from its members and blesses data for publication only after rigorous checking of provenance, calibration, and removal of all systematic effects. A member's good idea must be individually tested and then officially re-run by the group, commonly yielding turnaround times of weeks to months. Some experiments (e.g., Alice) process with a “data train,” or continuous data scan, with batch-like scheduling.

Non-HEP analyses are currently at smaller scales, though they are often more complex. They frequently require computations across related data items, whereas HEP analyses are based on statistics of (essentially) uncorrelated events.

3.10 80/20 rule

Most data-intensive users, including industry (e.g., Nokia, Facebook, eBay) and science, observed a common set of “80/20” characteristics although the exact numbers may vary:

a) less than 20% of data is accessed 80% of the time,
b) 20% of data changes all the time,
c) 20% of users consume 80% of the available resources.

Some industrial cases are more extreme. In one example, 70% of the users consumed only 2% of the available resources, so adding even a large number of this type of “simple reporting” user is still negligible. In all communities the hardest analytics were run by a very small number of experts; i.e., it is not unusual to see 2% of the users using 50% of the available resources. This disparity appears to be because the barriers to accessing and performing useful analyses on these large data sets are high.

3.11 Importance of raw data

A common theme among industrial and scientific users is the hunger for raw data. Industrial users have discovered the huge potential in mining a wide variety of logs (including website logs), which typically have been discarded. Sciences dealing with images expressed a strong interest in running complex analyses on the pixel data, not just derived data, and they would like to keep and analyze the pixels and their associated metadata in a single system.

Unfortunately, almost everybody is forced to throw away raw data. Sometimes this is justified. In HEP, only rare events are interesting – most events are well understood – but some fraction of those must still be discarded. Radio astronomy has plans to save only a small fraction of the incoming data. The biology community discards older raw data, since there is no desire to fund the archiving of the 5-6 PB generated each year, even though this makes it impossible to redo full analyses on previous samples. Industrial users discard some data (such as SMS messages) for legal reasons but still must discard other data due to cost.

3.12 Data ownership and usage

Virtually all scientists want to "own" and "control" their data[9]. They do not want *their* data to live on shared remote servers. But as their data sizes increase, their working sets may no longer fit on their desktops or laptops. The customary practice is to extract or derive subsets of the data and perform deeper or more specialized analysis on a local machine. Final analysis often involves specialized tools such as MATLAB, R, or their own code. In the past this local (offline) analysis mode was also often justified by poor or intermittent network connectivity, including the difficulty of accessing remote data while traveling. Yet as scientists attempt to use data more broadly and deeply, they are coming to accept the inevitable, frequent reliance on data servers; after all, finding the answers is more important than performance.

Pushing analyses into those servers can only succeed if the servers provide better support for scientists' familiar and currently local-only tools. Merely providing interfaces to download data does not address this issue. Also, the performance of the centralized systems should approach the performance of local analysis. Finally, ownership in a shared environment requires authentication and authorization controls to protect not only data but also code, letting users work privately without worrying about being "scooped" or letting others "steal" a Nobel prize. Centralized analysis requires a paradigm shift, but nobody doubts that it will be a huge win in the end.

3.13 Funding

Funding problems were discussed extensively at XLDB1 and XLDB2. At XLDB3, the geoscience representatives pointed out a disturbing trend: while data volumes have increased, the proportion of funding devoted to data management (DM) has decreased. Geoscience currently sees less than 10% of project budgets allocated for DM, whereas best estimates are that 30% could be needed to do an adequate job. One main reason is the idea that adding funds for DM "gets in the way of doing science" by reducing more direct science funding. While some centralized data managers such as the SDSS and Space Telescope Science Institute have demonstrated their value in terms of publications, in many other areas DM has struggled to prove its return on investment. The result is a "chicken-and-egg" problem – good DM-accelerated science results are difficult to produce without DM funding, but DM funding depends on the existence of those science results.

The user communities agreed that funding is somewhat misallocated, usually emphasizing the collection and production of data sets and neglecting their usage, analysis, hosting, or storage. Furthermore, the typically distributed nature of funding leads to the ownership challenges discussed in the previous section and makes it difficult to adopt potentially more-efficient centralized DM practices.

3.14 Inertia

The scientific and industrial communities differ greatly in terms of inertia, or resistance to change. Industry seems much more flexible, adjusting data models and tool sets rapidly, although perhaps still slower than desired. One participant noted: "Whatever we do, however agile we are, management asks for more and more and more." Scientists have a lot more inertia, preferring to avoid risk where the benefit (from DM) is difficult to measure, hence the prevalence of legacy software in science. Also, scientific environments are often centered on large collaborations whose use of tightly integrated tools and applications hamper change.

The ROOT system used by HEP is a good example of legacy, integrated software. It handles almost everything: data model and data storage, workflows, compression, schema evolution, histograms, statistics, and visualization. It contains much legacy code and is almost irreplaceable, for better or worse. Any new approach, such as map/reduce processing, must be carefully integrated with the ROOT format and framework.

[9] They often had to spend resources to obtain the raw data.

3.15 Other notes

3.15.1 Imbalanced systems

The XLDB3 participants worried that most systems funded and built for the science communities are poorly balanced for data-intensive scientific computing (DISC), instead being more suited for traditional high-performance computing (HPC). HPC systems are designed to process data, not move large quantities to and from storage, which may be perfect for running simulations, but bad for running large scale analytics.

When hardware is configured to provide the required I/O performance for DISC, software may become the bottleneck. When both CPU cores and disk bandwidth are plentiful, data management systems must use the available memory bandwidth well. Current systems often do not pay attention to this problem, as they are typically used in disk-bandwidth-limited configurations.

3.15.2 Cloud computing

Neither industrial nor scientific users currently rely on public clouds to store or analyze extremely large data sets. Science does not even use private clouds, using grids instead, which are much less elastic. The key obstacle may be the pricing model – for huge data sizes, private storage is cheaper than paying for bandwidth to and from the cloud and incurring monthly storage fees there. Yet there are signs of clouds in the future for analytics: the University of Washington's SciFlex system is an example, and there have been frequent inquiries whether SciDB will run on a cloud as a service. In the meantime, some are considering using a private cloud for bulk data processing and offloading to a public cloud during peak load.

3.15.3 Self-management

Everybody facing petabytes of data realizes the importance of auto-tuning. The sheer number of disks needed to manage petabytes and frequently changing hot spots means that manual administration efforts (e.g., for load balancing) require too many people and are thus too expensive in terms of labor. As one participant mused: “It's about having one or fewer DBAs [database administrators], who may be amateurs.”

3.15.4 Append-only

All agreed that petascale systems are (almost) all append-only – written once and never updated. There are many ways to take advantage of this feature to greatly optimize ingest throughput and improve concurrency.

3.15.5 Green computing

It is clear that the power costs of petascale computing are enormous. When many participants declared that electricity costs would soon exceed the purchase price of the computing hardware, others pointed out that this has already come true in some places. Hence petascale system design must consider power efficiency.

4 RDBMS VS. MAP/REDUCE

Many petascale users, scientific or industrial, have declined to adopt the RDBMS model for data management. The map/reduce model, on the other hand, has wide adoption within petascale industrial users and is undergoing preliminary testing (but not yet serious usage) among petascale scientific users[10]. This section discusses how these models differ and how these two paths appear to be converging.

4.1 Key differences

4.1.1 Procedural steps vs. monolithic query

The number one difference from an application programmer's perspective is the way data is accessed. In the map/reduce (MR) world, data is accessed by a pair of functions, one that “maps” all inputs independently and

[10] University of Nebraska-Lincoln and Caltech store simulated LHC/CMS data in Hadoop Distributed File System; anecdotal experimental usage by geoscientists and biologists; research usage by computer scientists.

one that “reduces” the results from the parallel invocations of the first. Problems can be broken down into a sequence of MR stages whose parallel components are explicit. In contrast, a DBMS forces programmers into less-natural, declarative thinking, giving them very little knowledge of or control over the flow of query execution. They must trust the query optimizer's prowess in “magically” transforming the query into a query plan. Compounding the difficulty is the optimizer's unpredictability: even one small change to a query can make its execution plan either efficient or painfully slow.

Some XLDB3 attendees reasoned that scientists, along with the engineers at Google/Facebook/Yahoo! types of companies, usually have PhDs and tend to think they can do better than an optimizer. They just “want the data and will deal with it.” Databases “get in the way.” Mathematicians and statisticians, who may have the most complex analytics, develop analytics procedurally and, unsurprisingly, favor the non-declarative map/reduce approach.

4.1.2 Checkpointing vs. performance

Map/Reduce systems perform frequent checkpointing: the outputs of each *map* step and each *reduce* step are saved, and users must express their tasks as sequences of MR stages. This checkpointing limits performance, but becomes critical in handling failures. In contrast, attendees pointed out that databases are built with the optimistic assumption that failures are rare: they generally checkpoint only when necessary due to resource limitations, e.g., when a sort/merge grouping algorithm is executed on a data set that won't fit into memory. Avoiding checkpointing leads to superior performance when there are no failures but much longer recovery time when failures occur. This has been shown through various studies[11]. At small scales, the assumptions made by databases are perfectly valid. In the petascale regime, thousands of disks often participate in answering a single query, making sophisticated fault-tolerance, perhaps even as far as re-observing lost data, increasingly necessary.

4.1.3 Flexibility and unstructured data support

The map/reduce paradigm treats a data set as a set of key-value pairs (sometimes with complicated values), much like an RDBMS has tables of tuples (relations). However, while the RDBMS model operates on sets, MR functions operate on a single pair at a time. The latter was thought to be more approachable for end users.

Data in databases are structured strictly in records according to well-defined schemata. Some participants noted that a database is “kind of like a prison,” and it can be a hassle to put data in and to take data out. MR is structure-agnostic, leaving interpretation to user code and thus handling both poorly-structured and highly-complex data. Loose constraints on data allow users to get to data more quickly, bypassing schema modeling, complicated performance tuning, and database administrators.

Relational databases require queries to be expressed in SQL, possibly with the addition of user-defined functions that operate through non-standard and non-portable interfaces. MR processing steps can be programmed in any language. Indeed, MR encourages users to employ their languages of choice, leveraging whatever libraries are available.

4.1.4 Cost

Finally, the last key difference is cost. High-end database system software is very expensive, and low-end alternatives require a lot of custom code on top to be usable in large-scale environments. There are no free-license, scalable (shared-nothing massively parallel) database systems on the market today. The maintenance cost of a large database setup is also non-negligible, and when database administrators are required, they can contribute significant overhead. At the same time, the MR world has Hadoop, a free open-source system. Hadoop's simplicity tends to lower the administration cost.

Unfortunately, MR jobs often lack optimizations employed by databases. Large tasks that can be executed on a small number of database nodes may require thousands of MR nodes to achieve comparable performance. For example, eBay employs a 96-node database cluster that routinely handles 70,000 queries per day over 6.5

[11] See e.g. A. Pavlo, E. Paulson, A. Rasin, D. J. Abadi, D. J. DeWitt, S. Madden, and M. Stonebraker, *A Comparison of Approaches to Large-Scale Data Analysis*, Proceedings of the 2009 ACM SIGMOD

petabytes of production data. The eBay team was told that a comparably performing MR cluster would require tens of thousands of nodes. Thus, in practice, MR has significant physical costs in hardware and operations (power and cooling).

4.2 Convergence

Despite their differences, the database and map/reduce communities are learning from each other and seem to be converging.

The map/reduce community has recognized that its system lacks built-in operators. Although nearly anything can be implemented in successive MR stages, there may be more efficient methods, and those methods do not need to be reinvented constantly. MR developers have also explored the addition of indexes, schemas, and other database-ish features[12]. Some are building a complete relational database system on top of MR[13].

Database systems are becoming more aligned with the map/reduce processing style in two ways:

1) Every parallel shared-nothing DBMS can use the map/reduce execution style for internal processing – while often including more-efficient execution plans for certain types of queries. Though systems such as Teradata or IBM's DB2 Parallel Edition have long supported this, a number of other vendors are building new shared-nothing-type systems[14]. It is worth noting that these databases typically use MR-style execution for aggregation queries.
2) Databases such as Greenplum and Aster Data (and soon, Teradata) have begun to explicitly support the map/reduce programming model with user-defined functions. DBMS experts have noted that supplying the MR programming model on top of an existing parallel flow engine is easy, but developing an efficient parallel flow engine is very hard. Hence it is easier for the DBMS community to build map/reduce than for the map/reduce community to add full DBMS functionality.

5 SOLUTION PROVIDERS

XLDB3 included unstructured talks from solution providers on various topics. This section describes what they presented.

All presenting solution providers were interested in working with the scientific community.

5.1 MonetDB

The MonetDB team has demonstrated significant interest in scientific data applications. Their successful port of the SDSS multi-terabyte database to MonetDB was remarkable in light of several other unsuccessful attempts to port SDSS data to other systems. They noted that the process required some changes to their SQL dialect for compatibility with SDSS's dialect such as adding a “top-k” function, removing 32-bit limits, adding custom spatial indexing, and adding support for UDFs. They changed the SDSS-provided queries by generalizing SQL Server-specific portions, in particular rewriting some spatial index C# code in pure SQL. Bulk data migration was facilitated by a special SQL Server-MonetDB connector.

Other notable aspects of MonetDB include its application of compiler optimization techniques to SQL, such as instruction parameter matching (common subexpression elimination) and instruction subsumption (strength reduction).

5.2 Cloudera

[12] An example of that is Hive, http://hadoop.apache.org/hive/

[13] HadoopDB http://db.cs.yale.edu/hadoopdb/hadoopdb.html

[14] ParAccel, Vertica, Aster Data, Greenplum, DATAllegro (now part of Microsoft), Dataupia, Exasol, SciDB, ...

Cloudera provides enterprise-level support for Hadoop, the popular open source implementation of map/reduce. They employ 15 of the 200+ Hadoop development team and are extending Hadoop to support columnar storage. Cloudera cited their experience working with scientific communities to ease adoption of the map/reduce model and expressed interest in continuing such activities.

5.3 Teradata

Teradata reported work in several areas that specifically address petascale needs. First, their automatic tuning and management features mean that no additional DBAs are needed as data scales up. They noted that poor query optimizations should be treated as bugs and not a DBA tuning task – no optimizer hints should be needed. Second, their storage virtualization layer for heterogeneous systems provides automatic migration of data to faster or slower storage based on “temperature” or frequency of access – hot data moves to fast storage and cold to slower/cheaper. Some participants noted that this technique is purely reactive and that human intervention can be proactive to help optimize the system for known future workloads.

5.4 Greenplum

Greenplum claimed to have a customer with the largest database installation – 6.5 petabytes on 96 nodes. They report one of the customer's biggest problems was the infrastructure proliferation problem: the maintenance of dozens of data marts and frequent new data mart births at a single site. Ideally, the system would support an “elastic infrastructure” in which a large pool of servers could be remapped to different warehouses dynamically and centrally.

Greenplum expects another release later this year to include columnar storage, external table support, additional indexing support, and parallel network ingest/export.

5.5 Astro-WISE

The Astronomical Wide-field Imaging System for Europe, or Astro-WISE, was built specifically to manage scientific data sets. Though astronomy-focused initially, it is now used for some non-astronomy projects. Astro-WISE stores pixel data in file servers and metadata in a database, mediating all I/O through the database. Instead of periodic data releases, Astro-WISE reprocesses data dynamically, which trades off access latency in favor of flexibility. Process-provenance tracking is facilitated by restricting processing to uploaded and registered executables, making this a comprehensive, integrated system somewhat like ROOT.

5.6 SciDB

The SciDB team built and demonstrated a from-scratch prototype system. Their system was notable for its n-dimensional array (with nesting) data model.

Their prototype stores data in compressed chunks distributed across multiple nodes. Queries were executed in parallel on heterogeneous nodes, using a scatter/gather technique for data redistribution. Their UDF model provides array-level access to cells. The demonstration illustrated scientific tasks such as object detection and regridding on raw pixel data and also showed a more relational filtering operation, confirming that tables map easily onto arrays. They pointed out the natural fit of gridded raw data to an array model, in contrast to a relational model where the overhead of storing dimensions would be prohibitive.

Future SciDB development will produce an alpha version more suitable for early-adopter experimentation and open development. The release will include source, binaries, and documentation and is scheduled for the end of March 2010.

6 SCIENCE BENCHMARK

6.1 Concept

XLDB2 introduced the concept of a science benchmark as a means to validate and compare solutions for large scale scientific analytics. The benchmark specification was to be derived from existing practice and would not be created just to demonstrate software capabilities. It was noted that the existing TPC benchmarks were overly complex and micro-optimized by vendors and therefore were no longer good examples.

Michael Stonebraker drafted the first specification. It covered raw data processing (pixel or gridded data processing), derived data analytics in an array model context, and was intended to simulate astronomy and geoscience usage. Specifically, the benchmark included “cooking” observations from raw images, matching observations between images, finding near neighbors, and time-series analysis.

6.2 Discussion

XLDB3 participants believed that the benchmark should be comprehensive, representing important large-data problems from a variety of science domains, but admitted this was very ambitious and would be best achieved in multiple, focused stages. They fully agreed that the array data model coverage was a very good approach as a first stage.

The current draft of the benchmark was mostly composed of descriptive text. Participants agreed that a releasable benchmark would need extra scaffolding, such as more precise mathematical requirements, rules for query execution, lists of acceptable and unacceptable configurations (e.g., permitted degree of replication), correct answers to queries, failure modes, and details on data loading. The benchmark also needed a review from a broader community of those facing array-data-related problems.

Participants suggested the formation of a dedicated working group for future benchmark development and identified a few candidates for that group. Many suggested that a “challenge paper” describing various scientific data analytics needs would be more appropriate as a starting point, given that the tasks themselves are not commonly understood in the petascale community. Such a paper would not need the detail and rigor that may be expected in a benchmark.

7 NEXT STEPS

7.1 XLDB's focus

Some participants suggested the scope of XLDB should be extended beyond “databases,” covering broader topics related to data management, such as file management, workflows, and pipelines. Others remarked that XLDB should “scale down to include people who are turned away or afraid of petabytes.” There was some disagreement over whether XLDB should be distinct from or tightly integrated with the SciDB project it spawned. The organizers will consider these debates and poll the community through the XLDB mailing list. The current thinking seems to be that it is better to stay tightly focused than to try to be more inclusive and risk slowing progress.

7.2 Reaching out

Each XLDB iteration has targeted different communities, yet there are still some that have never attended. These communities include oil and gas research, chemistry, medical informatics, banking and consumer credit, as well as the military. Geographically, Asia and the Southern Hemisphere have not yet participated significantly. Some communities (e.g., chemistry) may not have *extremely* large data sets, although they may experience complex issues in medium-scale data analytics.

7.3 Collecting use cases

The participants unanimously agreed that use cases from diverse science domains would best illuminate the needs of different communities. Industrial use cases are also useful although many come with secrecy constraints that limit their value. Representatives from biology, geoscience, and HEP agreed to provide their use cases within three months following this workshop. Collecting and publicizing use cases has been one of XLDB's most valuable services, and summarizing them into challenges/benchmarks would be “even better.”

7.4 Funding opportunities

XLDB3 participants discussed new funding opportunities from the European Commission. The EU has recognized the emergence of “big-data science” and has allocated new funds for “e-Science” and “e-Infrastructures.” The XLDB community was advised to consider writing a ~€200k “FP7-INTEGRATION” proposal encompassing activities such as finding common roadmaps, forging international partnerships, and organizing workshops and meetings. The proposal could provide seed money for a bigger (multi-million Euro) “FP7-INFRASTRUCTURES” proposal. FP7 proposals are intended for international software development for science, which would include development of a large-scale science analytics platform. Proposal acceptance requires participation from at least three large European scientific laboratories and strongly favors US collaboration. The European Commission works with US funding agencies in the event of such international efforts. This year's submission deadline is 24 November.

Participants noted that the preparation of such a proposal would be complex and non-trivial. Some believed that the presence of funding would change the nature of the workshop and its participants, citing cases where financial tensions have destroyed well-functioning communities. For these reasons, the XLDB3 organizers and others interested will perform appropriate research and exercise extra care before arranging what would be a complex, multinational funding structure.

7.5 Publicity

XLDB3 participants discussed how to publicize the petascale data problem. While means such as wikis, blogs, Twitter, or other social media may prove effective, they require dedicated evangelists. None of the participants could perform this role due to other responsibilities. Since the XLDB series has always operated on donated resources, dedicated funding may help[15]. The XLDB wiki, set up shortly after XLDB1, has not been active. Even some appropriate documents, such as collected use cases, have been posted elsewhere due to low interest in the wiki. Some suggested presenting XLDB at SC[16] although this would be difficult given XLDB's present as-needed donation funding situation.

7.6 XLDB4

It was decided that XLDB4 would be held in the United States about one year after XLDB3. Although XLDB2's recommendation was to reach out to Asia after Europe, participants agreed to maintain American momentum with a US venue. In particular, a Silicon Valley location would help attract industrial users for whom overseas travel is difficult. The vast majority of XLDB3 participants came for XLDB and not for the adjoining VLDB'09 conference. Returning attendees found that XLDB “gets better each year” and rated this instance the most interactive of all with the most productive, lively discussions.

Participants strongly believed that industry attendance was insufficient and should be increased. Vendor presentations were highly appreciated, and participants looked forward to hearing about other vendors' plans at future XLDB events.

[15] The entire XLDB effort so far has been executed through in-kind resources, plus help from sponsors whose funds were used towards funding catering at the workshops and associated social events.

[16] International Conference for High Performance Computing, Networking, Storage and Analysis

ACKNOWLEDGMENTS

- XLDB3 was sponsored by Microsoft, Teradata, and eBay.
- The organizers are grateful for the assistance of the local VLDB organizers, in particular Mohand-Said Hacid, for help with arranging the logistics in Lyon.
- The XLDB3 was organized by a committee consisting of Jacek Becla (SLAC, chair), Kian-Tat Lim (SLAC), Martin Kersten (CWI), Dirk Duellmann (CERN), and Maria Girone (CERN).

GLOSSARY

CERN – The European Organization for Nuclear Research
HEP – High Energy Physics
LHC – Large Hadron Collider
LSST – Large Synoptic Survey Telescope
netCDF – Network Common Data Form
PanSTARRS – Panoramic Survey Telescope & Rapid Response System
RDBMS – relational database management system
SDSS – Sloan Digital Sky Survey
SKA – Square Kilometer Array
UDF – user defined function
WAN – wide area network
XLDB – extremely large database

(Article history: Received 28 December 2009, Accepted 23 January 2010, Available online 14 February 2010)