Report from the XLDB-2015 Workshop

Jacek Becla and Kian-Tat Lim
SLAC National Accelerator Laboratory

The 8th Extremely Large Databases Workshop was convened alongside the XLDB-2015 Conference. This document summarizes the discussions that occurred during the workshop.

Large Cluster Deployment

Any XLDB solution requires deployment of software on a cluster of machines, whether locally or in the cloud. Companies operating at hyper-scale involving many thousands or hundreds of thousands of nodes routinely rely on containers to deploy their clusters. This contrasts with places managing smaller clusters (measured in tens or hundreds of nodes), which may use Chef, Puppet, or similar solutions to deploy their "small" clusters. The smaller practitioners may not have the expertise to build a container-based solution from scratch, however. It was agreed that it would be useful to document sample use cases from the “small cluster world” before the next XLDB event. A 1/2 day tutorial-type session featuring experts in the new container technologies to discuss these use cases and produce a set of recipes applicable for non-technical organizations with small clusters would be desirable. It has been noted that it is important to select hyper-scale experts with non-proprietary solutions.

Compute - Storage Decoupling

"Bring computation to data" has been a commonly-heard recommendation of luminaries and experts in extreme scale data management systems for many years. During XLDB-2015 we observed a new, somewhat conflicting trend: "decouple compute from storage". The main reason behind this trend is likely more pragmatic than technical: people value flexibility over absolute performance, and networks are becoming faster, making the flexibility increasingly attractive. This trend is particularly visible in systems with "big compute" on "smaller” (but still big) data. Data-intensive deep learning systems are one such example. Moving very large amounts of data across networks is still unlikely to be viable, but today it is harder to draw a clear distinction between when it is better to bring computation to data and when it is better to decouple compute from storage.

A debate involving proponents of both approaches at the next XLDB conference was suggested. The XLDB community should develop a taxonomy or set of scales characterizing the parameters of the problem space, such as the amount of raw data, the amount of data typically selected for computation, the compute to I/O ratio, and whether data is local to a server, a rack, a LAN, or not local at all. A useful outcome of such debate would be a clear engineering view of best practices as to what to use when, or perhaps simply a collection of "rules of thumb".

Data Rotting

Martin Kersten presented an enjoyable and stimulating informal talk on his "data rotting" concept. Going beyond multi-temperature freshness models for hierarchical storage, he suggested permanently removing data elements with low value in order to match the growth rate of data with the growth rate of storage. Data "fungi" would algorithmically determine which data to continuously "rot away". Various species of fungi could exist: simply removing data randomly, reducing data resolution, replacing data with (lossy) summaries, etc. Users would have to explicitly acknowledge the costs of keeping data by counteracting the removal with "medicine", if desired.

The discussion countered with several points. Old data may suddenly increase in value when new analysis methods are developed. Reproducing a computation of deleted data from raw sources may be difficult. The exponential growth of data means that old data is often insignificant in size compared to new data. Some users filter most of the data before it is ever stored, with the Large Hadron Collider at CERN being the prime example. But this requires being quite certain as to what to throw away.

This could be an interesting area for further academic research.

Late Schema Binding

It has been noted that systems involving late schema binding such as Splunk are becoming increasingly popular. Schema-less key-value type systems are in the same category. Part of the appeal of Splunk-like systems in particular may be the use of a sequence-based data model rather than the relational set-based model. A move towards such systems clearly is not driven just by curiosity; these systems are so convenient that they are finding real applicability in a wide range of applications every day, and people are getting a lot of value out of them.

The question was raised as to how significant and disruptive this trend really is. For some, this trend resembled the early days of the Hadoop ecosystem, where the database community considered Map/Reduce and Hadoop to be "rediscovering the wheel" and "unlikely to succeed".

A follow-up question was raised as to whether we can we get ahead of the train, see what barriers these systems are likely to run into, and provide early assistance.

Data governance was felt to be the key, regardless of the underlying technologies. Such governance includes business rules around the structure of data or lack thereof and the definitions of data items (including things such as units). Even with an RDBMS, much of this occurs outside the core engine. Defining the amount of "sloppiness" allowed in various analyses and reports is also important; providing less accurate data faster can be useful, and customers can understand what the data means. This contrasts with normal data warehouse practices where no results are provided unless they are guaranteed to be correct.

We're considering inviting experts representing a wide range of systems (Splunk/Hadoop/key-value/RDBMS) to the next XLDB, introducing the topic through a 30 min talk by e.g. a Splunk expert, followed by a discussion.

Integrating Scientific Analytics Into Databases

Scientific communities are struggling with integrating their specialized analytics with databases. This integration is desirable both because it can provide data management features and because it can enable high-performance execution of operations, including parallel execution on distributed data.

Types of integrations include (1) putting scientific code inside a database through user-defined functions or similar mechanisms, (2) having scientific code transparently utilize a database for data storage and appropriate operations through mechanisms like R extensions, or (3) the traditional but usually inefficient method of extracting data using a database's API, operating on it with scientific code, and returning results to the database.

The need to repackage or rewrite legacy software is a major obstacle to the first type of integration, particularly when the code needs to run in a parallel environment. The second type of integration is showing promise for low-level operations, but the best performance will only be available when the language or scientific application and the database can work together to optimize more globally.

Next Conference

Based on input from the community during and after the conference, we should keep the event in Silicon Valley and keep it as annual event. The fact that almost all speakers this year were from the SF Bay Area underlines the importance of location. To reduce the load on the SLAC team, we will start organizing the next event much sooner than before, and we will attempt to engage the XLDB community more. In particular, we made great progress with identifying topics for the next event during the workshop; once the topics are further fine-tuned, we will identify moderators for each block of the conference and will delegate to them the work related to coordinating their block of program. Further, we created an opt-in Google working group (http://groups.google.com/d/forum/xldb-discuss) to discuss both technical and organizational topics related to XLDB without burdening the XLDB-L mailing list. To publicize the event (and avoid getting lost in the sea of big data events), we will be promoting the event through personal connections (word of mouth), and using existing channels such as Twitter and LinkedIn. This approach was deemed most effective given we want to attract specific groups of specialists. Jeremy Cole agreed to lead the PR effort on behalf of the XLDB community. We will be also exploring whether there are any collaboration opportunities with the Stanford marketing department.

Some possible topics for the next conference:

- decoupling of compute and storage (and even memory)
- real time analytics (Uber decision making system, stream processing focused on data reduction, what to log / what not)
- reproducibility, repeatability
- long term storage and the utility of tape, disk, flash, and other storage technologies
- security in XLDB systems
- late-binding as an emerging trend
- add more science-related speakers.

The XLDB-for-government event will likely be organized early next spring as a satellite event.

The topic of documenting and disseminating information from XLDB meetings was discussed. While it would be very useful to develop guidelines / cookbooks / best practices manuals, it is a complex multi-dimensional space, and the world is moving so fast that it was considered too difficult and impractical. The existing method of disseminating XLDB information (YouTube videos) was considered "already much better than most conferences" and sufficient.