Bloor
InBrief

Philip Howard – Research Director, Information Management

Bloor MUTABLE AWARD 2019 ONE TO WATCH

LeanXcale

www.LeanXcale.com

CAIT, 28223, Madrid
Tel: +34 656 68 29 48
Email: info@leancale.com

The company

LeanXcale is based on research conducted at the Distributed Systems Laboratory within Madrid Technical University (UPM). Initial development was funded through EU R&D grants and a prototype was first delivered in 2010 with the company being formally founded in March 2015. The company has patents (both in Europe and USA) for both scaling query processing and ultra-scalable transaction management.

LeanXcale has its headquarters in Madrid as well as an office in the United States. While the LeanXcale database is suitable for deployment in any OLTP or hybrid OLTP/OLAP environment, the company is primarily focused on financial services, telecommunications, retail and machine-to-machine environments.

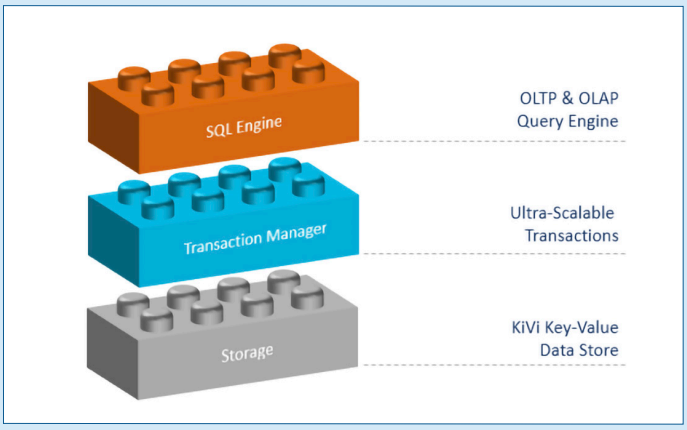


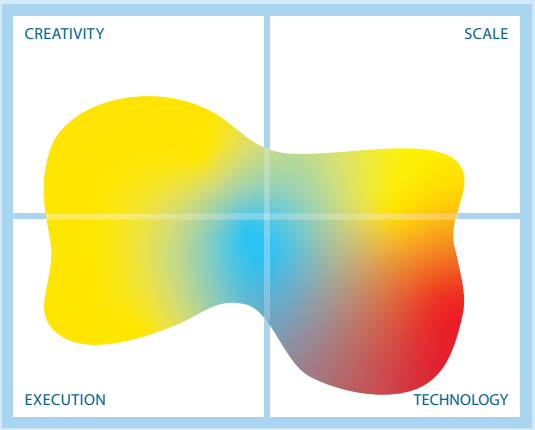
Figure 1 – The three-tier architecture

What is it?

LeanXcale is an ACID compliant, shared nothing, NewSQL database that has a three-tier architecture, as illustrated in ***Figure 1***. The storage engine (KiVi) is a relational key-value store that also provides the option to persist data in column format. Compression techniques against columnar data are currently under development.

The Transaction Manager has multiple sub-components (as do the other elements of this architecture), which each specialise in one of the ACID properties. In effect, each of those properties scales independently. In particular, one of the company's patents applies to how immediate consistency is supported across the distributed, scale-out architecture that LeanXcale provides. In this context, it is worth noting that LeanXcale does not employ explicit sharding, thus reducing the cost and complexity of development. Instead, there is internal auto-sharding that is applied transparently to the applications.

The SQL engine does what its name suggests. However, it is not limited to use with LeanXcale itself. The product also supports federated polyglot query capability with other SQL databases and NoSQL data stores such as MongoDB and HBase. In addition to SQL you can also use native key-value access to the data. This is especially useful for ingesting data because it bypasses the overhead that using SQL introduces. LeanXcale also integrates with streaming platforms such as Spark and Flink.



The image in this Mutable Quadrant is derived from 13 high level metrics, the more the image covers a section the better. Execution metrics relate to the company, Technology to the product, Creativity to both technical and business innovation and Scale covers the potential business and market impact.

How does it work?

LeanXcale is both vectorised and employs SIMD (single instruction multiple data) processing. It supports secondary indexes and has developed its own technology for primary indexes which, the

Architecture	4.5	Flexibility	3.5
Concurrent analytics	4.5	In-process analytics	2.5
Criticality	4	Performance	4.5
Ease of use	4	Scale	4

> "The product is the raesult of 15 years of research and it shows: the company has significantly re-thought how a database to process transactions, with or without simultaneous query processing, should be designed."

company claims, combines the benefits of both B+ trees and LSM (log-structured merge) trees. As mentioned, there is no explicit sharding (the manager performs auto-sharding transparently to the application) but the product should scale (elastic scaling is provided) to hundreds of nodes and internal benchmarks (TPC-C and TPC-H like, and YCSB) suggest that the product scales out linearly. Workload management capabilities and dynamic load balancing using an algorithm developed by the company, to guarantee uniform resource usage that enables the linear scale-out. Active-active replication is provided to support high availability and there is another company-developed algorithm to support geo-replication.

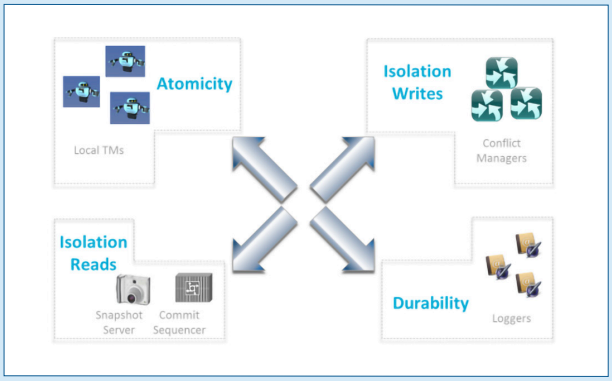


Figure 2 – Scaling ACID properties

While there are several secret sauces behind LeanXcale, perhaps the most interesting is the way that the product scales ACID properties. As noted previously, each of these is addressed individually, as illustrated in ***Figure 2***, but the key point is that LeanXcale processes transactions in parallel across multiple nodes (where traditional approaches have a single node bottleneck) while maintaining a consistent view across those transactions. The main principles behind enabling this are that the commit processes themselves are separated from the visibility of the committed data, that timestamps are pre-assigned to committing transactions (so that serialization order is defined), and the detection and resolution of conflicts prior to commit.

While there are multiple elements supporting this approach, the three most worthy of discussion are the Snapshot Server, Commit Sequencer, the Conflict Managers and the Loggers. The first two of these supports the product's snapshot isolation, and which keeps track of the most recent transaction that is consistent.

Each Conflict Manager takes care of a subset of keys and there can be as many as you like. In principle, maintaining concurrency control in this way should be much more efficient for batching purposes. For example, when LeanXcale ran (internally) a TPC-C benchmark, it achieved a twenty times performance boost for batching.

Each Logger takes care of a fraction of the log records and scale in the same way that Conflict Managers do. They run in parallel and can be replicated. The upshot of this approach is that LeanXcale supports multiple durability models: replicated memory, 1-safe and n-safe durability.

Finally, it is worth commenting on the way that query processing works. SQL is translated into a query plan that is represented as a tree of algebraic operators, which themselves are represented in the query engine as Java bytecode. At the leaves of the query plan there are Scan Operators that push down to KiVi predicate filtering, aggregation, grouping and sorting capabilities, that is, all algebraic operators below a join are pushed down.

Why should you care?

The preceding description does not really do sufficient justice to LeanXcale. We do not have the space to discuss the technology provided by LeanXcale in the depth it deserves. The product is the result of 15 years of research and it shows: the company has significantly re-thought how a database to process transactions, with or without simultaneous query processing, should be designed.

The Bottom Line

We are very impressed with LeanXcale. Neither the product nor the company are especially well-known, but they deserve much greater recognition.

FOR FURTHER INFORMATION AND RESEARCH CLICK HERE