

**Ultra-Scalable Operational Full SQL Full
ACID Database with HTAP Capabilities**

Table of content

LeanXscale: Ultra-Scalable Operational Full SQL Full ACID Database with HTAP Capabilities	3
The LeanXscale DBMS.....	3
Blending OLTP and OLAP functionalities	5
Architecture	6
Principles for Scaling	7
Principles for Efficiency	11
References	12

LeanXcale: Ultra-Scalable Operational Full SQL Full ACID Database with HTAP Capabilities

This whitepaper provides an overview of the LeanXcale platform, its design rationale, architecture and key distinguishing features.

The LeanXcale DBMS

LeanXcale is a full SQL full ACID database with Hybrid Transaction and Analytical Processing (HTAP) capabilities. On the operational side, it can scale to millions of update transactions per second thanks to its patented technology for scaling transactional management. LeanXcale addresses most of the needs for data management (see Figure 1). Let us examine each of the facets separately.

LeanXcale is an ultra-scalable **full ACID full SQL** database. One of the main distinctive features of the LeanXcale DBMS is that it can **scale from 1 node to 100s of nodes** while being able to process very large update transaction rates (i.e. many millions per second). It is based on a radically new approach to transactional processing that provides full transactional data consistency while being able to scale-out to large cluster sizes. **Its scalability is totally transparent to underlying applications:**



Figure 1: LeanXcale Ultra-Scalable OLTP database

- 1) Syntactically: since the database is accessed by means of a standard JDBC driver and Standard SQL and
- 2) Semantically: because it exhibits the same behavior and properties as any full ACID database.

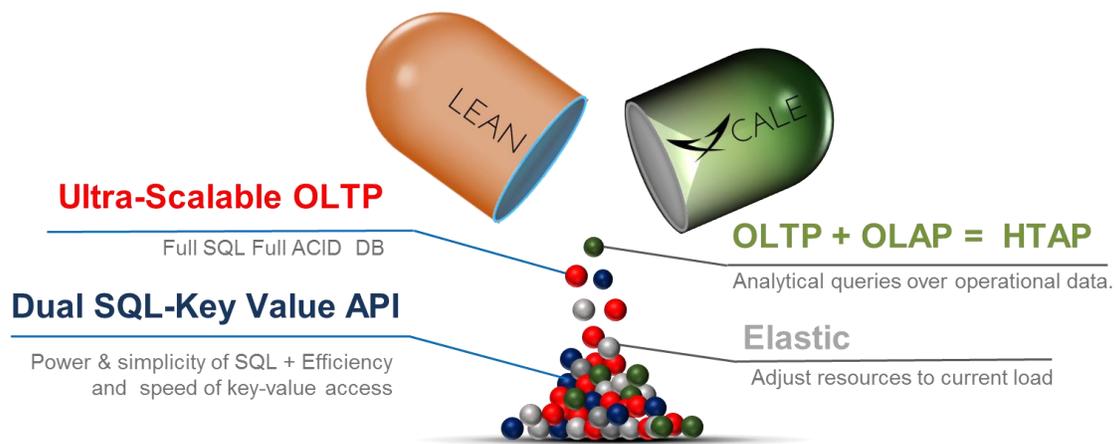


Figure 1: LeanXscale Data Management

Companies are exposed today to a myriad of technologies that address subsets of their data management needs that they struggle to coherently and efficiently integrate. This is nothing new and has been experienced in the past with the adoption of operational databases and data warehouses. In these scenarios data needs to be copied from the operational database into the data warehouse by means of ETLs (Extract-Transform-Load). A lot of engineering hours were spent creating ETL pipelines between the operational databases and data warehouses. Investment in operations teams to maintain the ETL, infrastructure and software estate was considerable and the architecture itself tended to be fragile with many moving parts requiring much attention to ensure daily business analytics.

Blending OLTP and OLAP functionalities

LeanXscale is equipped with a powerful Online Analytical Processing (OLAP) engine. The OLAP engine parallelizes queries over multiple nodes to answer heavy analytical queries in online response times. It is based on a specialization of a patented process for processing queries in a streaming manner [QueryPatent]. Queries aggregate the power of multiple servers to rapidly answer analytical queries. **The OLAP engine works over the operational data**, this is made possible because of the ultra-scalability of the transactional processing. This combination of cutting edge technologies enables LeanXscale to deliver **real-time analytics capabilities** reducing the number of ETLs needed or eliminating them altogether. Operational data can be kept in LeanXscale and can also be queried with pure analytical requests avoiding the need for traditional ETLs from operational databases to data warehouses. In today's modern architectures ETLs are estimated to account for 80% of the cost of current business analytics.



(a) Before LeanXscale: Siloed operational DB and DW requiring ETLs

LeanXcale
OLTP + OLAP = HTAP



Analytical Queries
on Operational Data

(b) After LeanXcale: Single DB for both operational and analytical workloads

Figure 2: Blending OLTP and OLAP

Architecture

LeanXcale is a shared-nothing distributed database that runs on commodity hardware either on-premise or in the cloud. Figure 3 depicts the architecture of LeanXcale in terms of subsystems. At the bottom, there is LeanXcale own proprietary storage engine known as KiVi. KiVi is an ultra-efficient distributed storage engine.

KiVi is itself a scale-out distributed relational and key-value data store. In fact, LeanXcale relational tables can be accessed from both the SQL interface and the key-value interface. This dual interface is really convenient because data can be efficiently ingested at very high rates with very little resources by using the key-value interface. Also, simple range queries

with predicates can be answered directly through the key-value interface in a very inexpensive way. KiVi is integrated with our ultra-scalable transactional manager, what means that is a fully ACID key-value data store.

On top of this, we have our distributed SQL query engine. The SQL query engine enables access to the data using SQL. It provides full SQL and a JDBC driver to access the database.

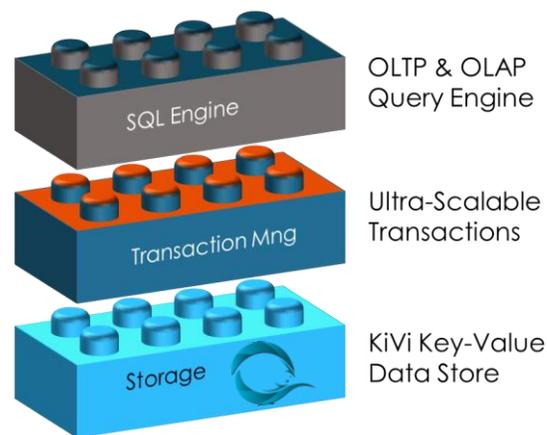


Figure 3: Architecture of LeanXcale Platform

Principles for Scaling

Traditional and modern transactional databases do not scale or at best scale in a very limited way because at some point of the transactional processing they perform some action(s) in a sequential manner that inherently create a single node bottleneck, or they have a centralized component that performs the transactional processing.

LeanXcale is able to scale-out to 100s of nodes thanks to its radical patented technology to transactional processing [TxnPatent] called Iguazu. With this patented method transactions are processed and committed fully in parallel without coordination across them.

Consistency is achieved by making concurrent snapshots of committed data visible as soon as they become consistent instead of delaying the execution or forcing sequential processing. In other words, instead of executing transactions sequentially or centrally to enforce consistency, transactions are allowed to execute fully in parallel and whenever a new consistent snapshot of the data is available it is then made visible to applications.

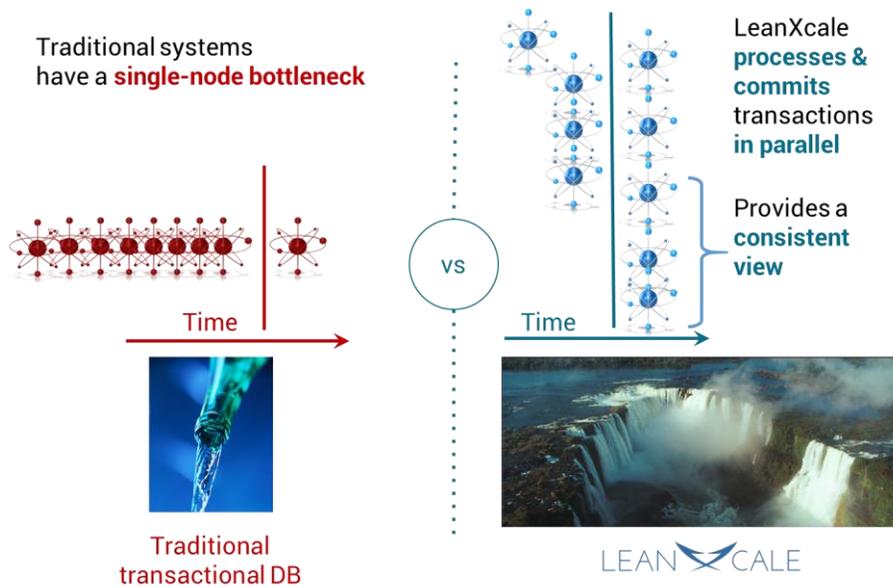


Figure 4: Traditional transactional databases vs. LeanXcale

LeanXcale in order to scale, avoided a centralized transactional manager that would become a single-node bottleneck and decomposed the ACID properties into its basic components.

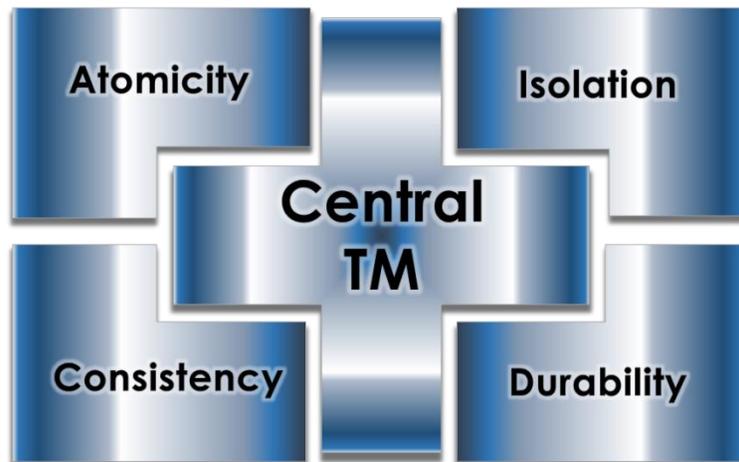


Figure 5: Central Transactional Manager providing all the ACID properties.

A transactional database has to care about Atomicity, Isolation, and Durability. Consistency, in the context of the ACID properties, is the responsibility of the application to transform a consistent state of the database into another consistent state, that is, the application should be correct and not introduce inconsistencies into a consistent database. What typically is understood as consistency is the combination of the other three properties. LeanXscale further decomposes isolation into the isolation of reads and isolation of writes, since they require different mechanisms. It scales out each of the properties individually in such a way that the combination provides ultra-scalable full ACID transactions.

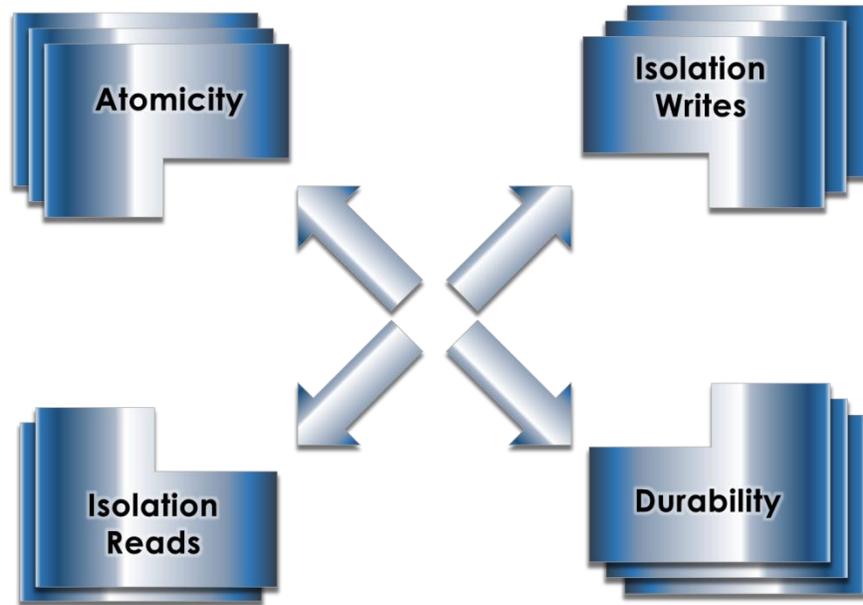


Figure 6: LeanXcale Decomposition of the ACID Properties and Scale Out of them.

LeanXcale uses the components depicted in xx for delivering each of the ACID properties in a scalable manner.

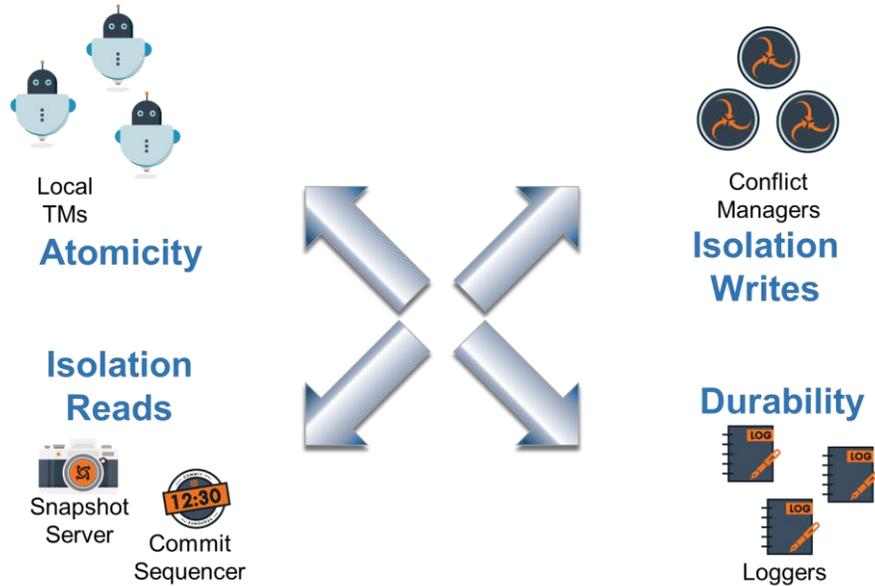


Figure 7: LeanXcale Transactional Management Components

Principles for Efficiency

An important feature of LeanXcale transactional processing is that it has been conceived to scale to very high transaction rates but also to scale efficiently. **LeanXcale is highly efficient at scaling transactions** thanks to its transactional processing that enables transaction batching systematically and efficiently processes thousands of items or transactions within a single message, vastly reducing the distribution overhead.

Existing storage engines waste many CPU cycles due to the fact they were originally designed for a single CPU world from decades ago. KiVi storage engine has an innovative architecture designed to run with very high levels of efficiency on modern multi-core and many-core architectures that follow a NUMA (Non-Uniform Memory Access) approach.

KiVi's architecture leverages this modern hardware avoiding the inefficiencies of older storage engines.

References

[TxnPatent] Ricardo Jimenez-Peris, Marta Patiño-Martinez. System and method for highly scalable decentralized and low contention transactional processing. Filed 18th Nov 2011, Published 23rd May 2013. Granted in the US and Europe. EP 2780832, US20160179876, WO/2013/072451

[QueryPatent] Ricardo Jimenez-Peris, Marta Patiño-Martinez. Parallel Processing of Continuous Queries on Data Streams. Filed at USPTO with application number 61/356,353 on 18th June 2010, Published, Granted in US and Europe. EP 2583175, US 13/805,307 (intention to grant given, waiting for the granted patent number), WO/2011/157442

Resources

Visit www.leanxcale.com for more information, or contact us at info@leanxcale.com.

Free Trial (<https://www.leanxcale.com/trial>)

Documentation and drivers (<https://www.leanxcale.com/company-resources>)

Whitepapers and videos (<https://www.leanxcale.com/company-resources>)

Get a demo (<https://www.leanxcale.com/get-a-demo>)

Talks (<https://www.leanxcale.com/talks>)

Blog (<https://www.leanxcale.com/blog>)