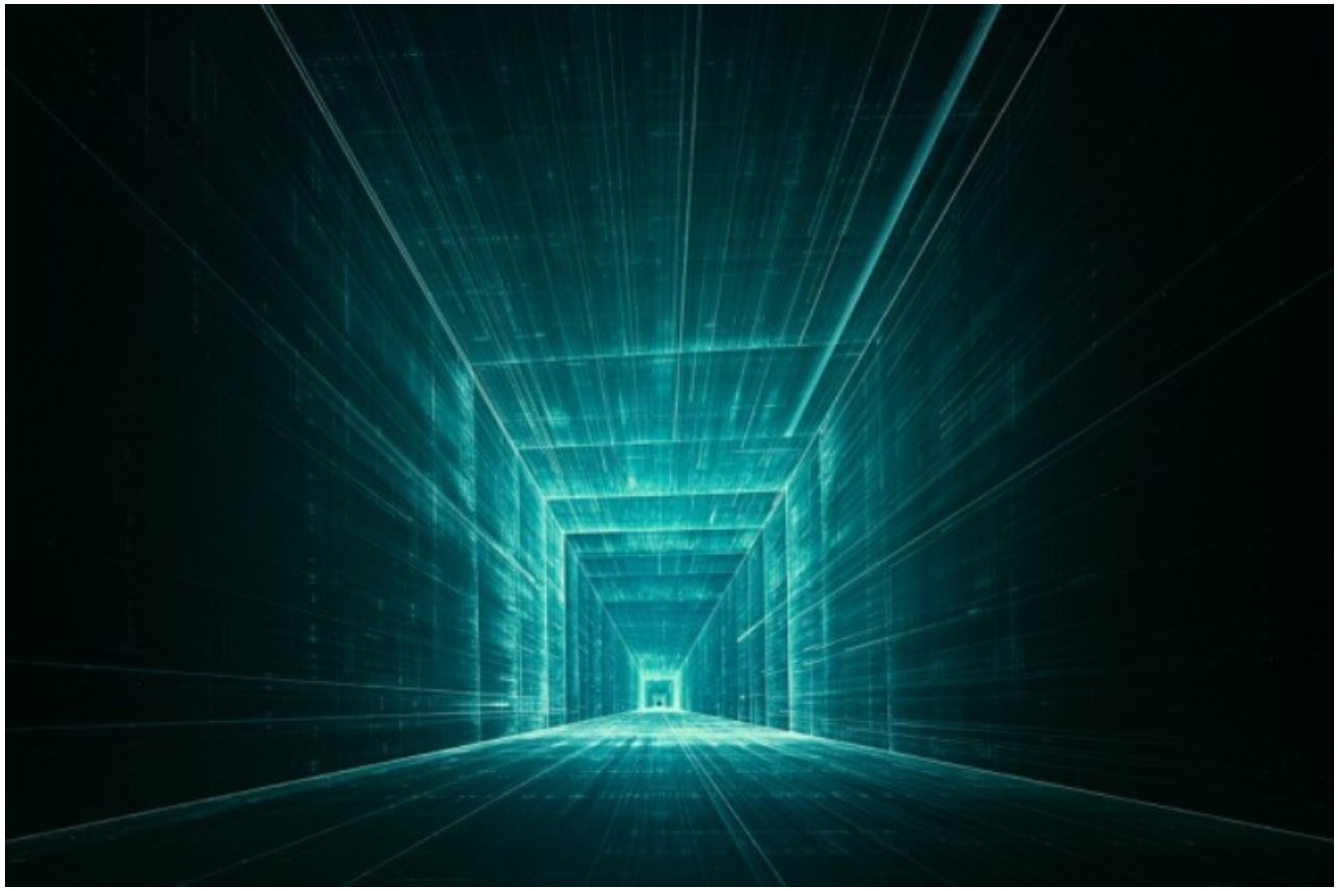
Taxonomy of NewSQL Systems



SQL versus NoSQL versus NewSQL

	SQL	NoSQL	NewSQL
Data model	Relational	Specialized: KV, document, graph	Relational
Query language	SQL	Key-value API New query language	SQL Key-value API
Scalability	Only for high-end (Teradata, Exadata)	By design (SN cluster)	By design (SN cluster)
Consistency	Strong	Limited	Strong
Big data ecosystem	External tables (HDFS)	Integration within Hadoop	Integration within Hadoop
Workload	OLAP or OLTP	OLTP	OLAP, OLTP, HTAP
Polystore	SQL or HDFS data sources	SQL or HDFS data sources	SQL, NoSQL, HDFS data sources

Taxonomy Definition

- Why a taxonomy?
  - Different flavors of NewSQL systems, for different workloads
- Main dimensions
  - Transaction model
  - Scalability
  - Storage engine
  - Query parallelism
  - Polystore

Transaction Model

- Ad hoc
  - ACID properties partially provided
  - Isolation and atomicity not guaranteed
- A priori-knowledge
  - Requires to know which rows will be read and written before executing the transaction
- ACID
  - ACID properties fully provided
  - Isolation levels well such as serializability or snapshot isolation

Scalability

- Bounded scalability
  - Centralized transaction manager: can scale out as far as it does not get overloaded
- Logarithmic scalability
  - ROWA replication: can scale out the read workload
  - Cache consistency: requires synchronization of the updated blocks
  - 2PC: to deal with multi-node transactions
- Linear scalability
  - Linear scale-out in shared-nothing cluster

Storage Engine

- Relational
  - Support of algebraic operators, such as predicate filtering, aggregation, grouping and sorting
- Read/write, key-value
  - Capability of reading and writing individual data items
  - May support range queries

Query Parallelism

- Inter-query parallelism
  - Multiple queries can be processed in parallel
- Intra-query parallelism
  - Inter-operator parallelism: different operators in the query plan can be processed on different nodes, but a single operator runs on a single node.
  - Intra-operator parallelism: the same operator can run across multiple nodes

Different Flavors of NewSQL Systems

- SQL+key-value
  - Use a key-value data store to scale data management, e.g. Splicemachine, EsgynDB and LeanXcale.
- HTAP
  - Systems that combine OLTP and OLAP, e.g. SAP Hana, EsgynDB and LeanXcale.
- In memory
  - Optimized for processing the workload fully in main memory, e.g. SAP Hana and MemSQL
- New transaction managers
  - New, scalable approach to transaction management, e.g. Spanner and LeanXcale

Some NewSQL Systems

Vendor	Product	Objective	Comment
Google	Spanner	OLTP	Google cloud distributed database service. Used by F1 for the AdWords app.
LeanXcale	LeanXcale	HTAP	HTAP DBMS with fast insertion, fast aggregation over real-time data and polystore capability
SAP	Hana	HTAP	The HTAP pioneer, based on in-memory, column store
MemSQL Inc.	MemSQL	HTAP	In-memory, column and row store, MySQL compatible
Esgyn	EsgynDB	HTAP	Apache Trafodion for OLTP, Hadoop for OLAP
NuoDB	NuoDB	OLTP	Distributed SQL DBMS with P2P architecture
Splice Machine	Splice Machine	HTAP	HBase as storage engine, Derby as OLTP query engine and SparkQL as OLAP query engine

Google Spanner

- Globally distributed database service in Google Cloud
  Synchronous replication between data centers with Paxos
  - Load balancing between Spanner servers
    - Favor the geographical zone of the client
- Different levels of consistency
  - ACID transactions
  - *Snapshot* (read only) transactions
    - Based on data versioning
  - Optimistic transactions (read without locking, then write)
    - Validation phase to detect conflicts and abort conflicting transactions
- Two interfaces
  - SQL
  - NoSQL key-value interface
- Hierarchical relational storage
  - Precomputed joins

LeanXcale

- SQL DBMS
  - Access from a JDBC driver
  - Polyglot language with JSON support (pending)
  - SQL processing based on Apache Calcite
- Key-value store (KiVi)
  - Fast, parallel data ingestion
  - Online aggregation
  - Multistore access: HDFS, MongoDB, Hbase, …
- OLAP parallel processing (Query Engine)
  - Intra-query intra-operator parallelism
- Ultra-scalable transaction processing (patented)
  - SQL isolation level: snapshot
  - Timestamp-based ordering and conflict detection just before commit
  - Massive parallel commits of transactions

Comparisons

System	Trans. model	Scalability	Storage	Query parallelism	Polystore
Spanner	ACID	Linear	Key-value	Inter-query	
LeanXcale	ACID	Linear	Relational key-value	Intra-query Intra-operator	SQL, NoSQL, HDFS
Hana	ACID	2PC	Relational Columnar	Intra-query Intra-operator	HDFS
MemSQL	Ad hoc	Log	Relational	Inter-query	HDFS (Spark connector)
EsgynDB	ACID	2PC	Key-value	Intra-query Intra-operator	HDFS
NuoDB	ACID	Log	Read/write	Inter-query	HDFS
Splice Machine	ACID	Centralized TM	Key-value	Intra-query Intra-operator	HBase, HDFS

Conclusion

- NewSQL is the hottest trend in database management
- Scales out but without renouncing to SQL and ACID transactions
- NewSQL has different roots leading to different flavours
- The taxonomy enables comparing NewSQL systems as well as SQL systems

Current Trends

Business Perspectives

- NewSQL is the database kind growing faster in the market
  - 33% CAGR, according to 451 Research market analyst in their Total Data Report
  - It will become soon an important player in the database landscape
- Early adopters of NewSQL will become leaders in database management
  - Faster database development with lower costs in engineering and lower needs in talent

Many Research Opportunities

- Polyglot SQL
  - SQL++ compatibility
  - JSON indexing within columns
- Polystore
  - Cost model, including histograms
  - Materialized views
- Streaming and CEP
  - Query language combining streaming and access to the database, e.g., through SQL or KiVi API
- Scientific applications
  - NewSQL/HTAP + scientific workflows
- Analytics and ML
  - Spark ML using updatable RDDs, instead of redoing RDDs periodically,
  - Incremental ML algorithms based on online aggregation, scalable updates and OLAP queries
- Benchmarking
  - Defining NewSQL/HTAP benchmarks and compare systems
  - Profiling to find new optimizations

References

1. T. Özsu, P. Valduriez. *Principles of Distributed Database Systems*. Fourth Edition. Springer, 2020.

2. P. Valduriez, R. Jimenez-Peris. NewSQL : principles, systems and current trends. *IEEE BigData*, Los Angeles, December 2019.

3. R. Jimenez-Peris, M. Patiño-Martinez. System and method for highly scalable decentralized and low contention transactional processing. Filed at USPTO: 2011. European Patent #EP2780832, US Patent #US9,760,597.

4. C. Bondiombouy, P. Valduriez. Query Processing in Cloud Multistore Systems: an overview. Int. Journal of Cloud Computing, 5(4): 309-346, 2016

5. B. Kolev, O. Levchenko, E. Pacitti, P. Valduriez, R. Vilaça, R. Gonçalves, R. Jiménez-Peris, P. Kranas. Parallel Polyglot Query Processing on Heterogeneous Cloud Data Stores with LeanXcale. *IEEE BigData*, 2018.

6. B. Kolev, P. Valduriez, C. Bondiombouy, R. Jiménez-Peris, R. Pau, J. Pereira. CloudMdsQL: Querying Heterogeneous Cloud Data Stores with a Common Language. *Distributed and Parallel Databases*, 34(4): 463-503, 2016.

7. B. Kolev, C. Bondiombouy, P.Valduriez, R. Jiménez-Peris, R. Pau, J. Pereira. The CloudMdsQL Multistore System. *ACM SIGMOD,* 2016.

8. M. Stonebraker.The Case for Polystores. ACM blog, July 2015.