Cloud Native Transactions and Analytics in SingleStore

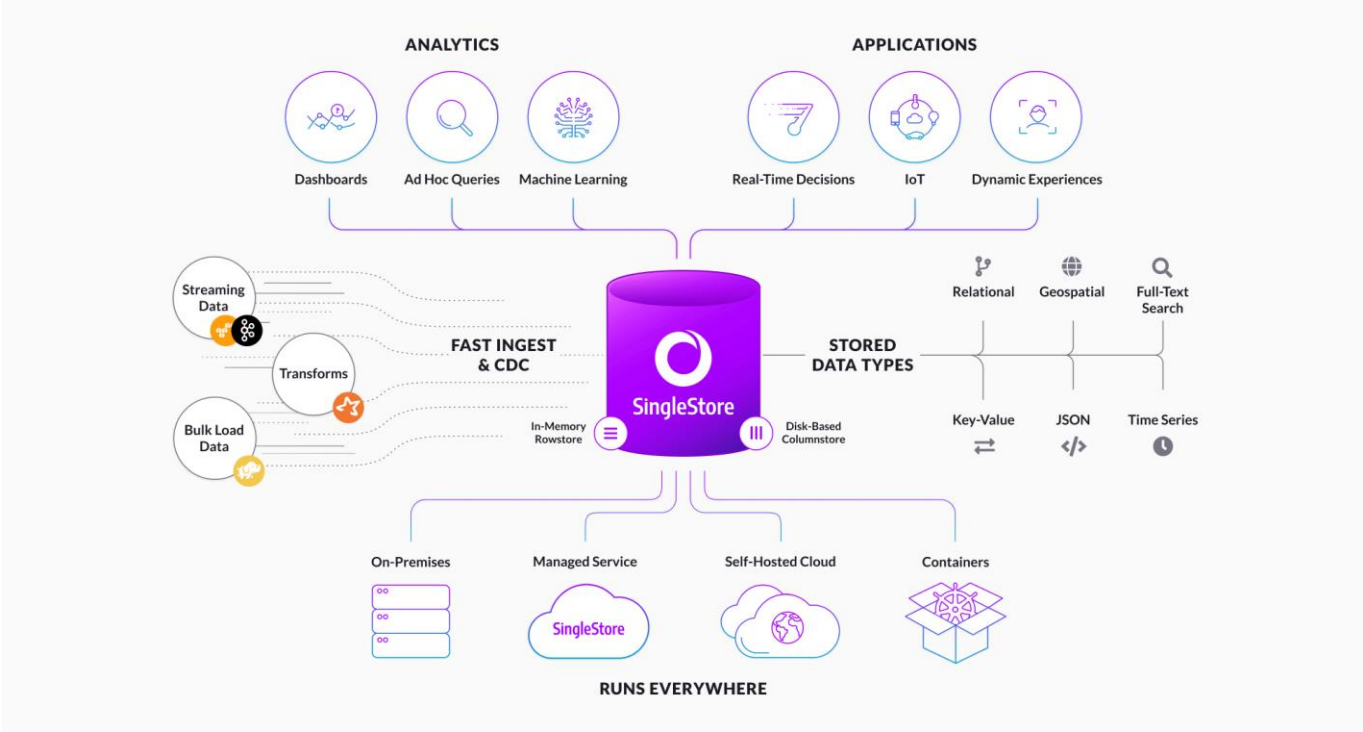
Madhushree Nijagal

Motivation

- Complexity and Cost of multiple specialized systems in a single application environment
- Need for a unified data platform for OLTP and OLAP transactions
- Highly available, durable storage and elastic compute instances
- Store more data and access with lower latency and high throughput

Goals

- Build a fast, distributed general purpose SQL database
- Unified database that handles transactional and analytical workloads with strong performance
- Support a breadth of workload over disaggregated storage



Background

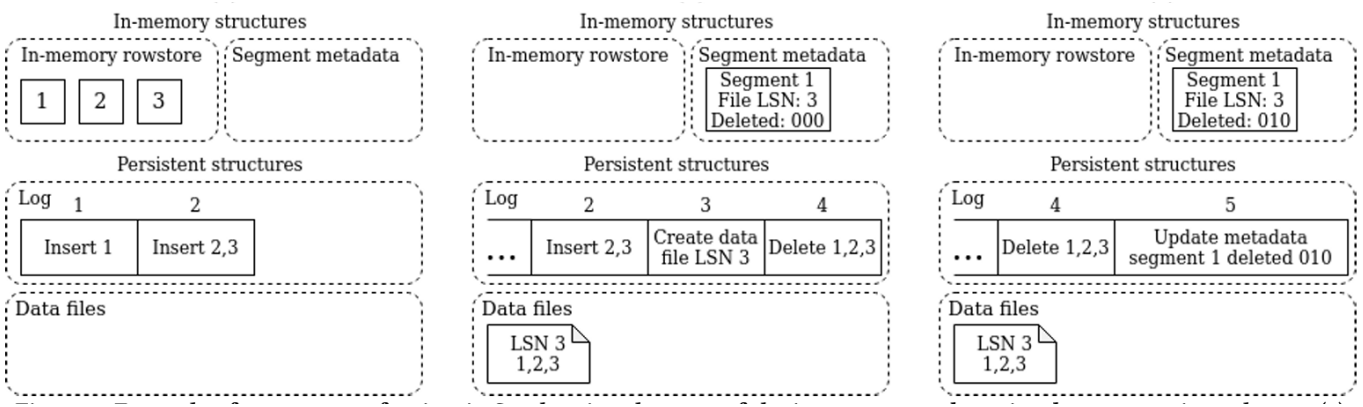
- Horizontally partitioned shared-nothing DBMS
- S2DB cluster is made up of aggregator nodes and leaf nodes – holds partitions of data
  - Aggregator nodes responsible for coordination of queries
  - Leaf nodes responsible for compute of queries – master(reads/writes) and replicas (reads)
  - Stores multiple replicas of each partition on different nodes in a cluster
- Uses LLVM for full query code generation in contrast to hand built interpreters

Components of S2DB

- Separation of storage and compute
- Unified table storage

Table storage format

- Rowstore storage
  - In memory rowstore table uses lockfree skiplist
  - Readers don’t wait on writers
  - Writers use pessimistic concurrency control
  - Log created for each database partition and persisted to disk
- Columnstore storage
  - Data stored in segments as an LSM tree stored on disk as data files
  - Common encodings – bit packing, dictionary, run length encoding used on columns
  - Segment metadata stored

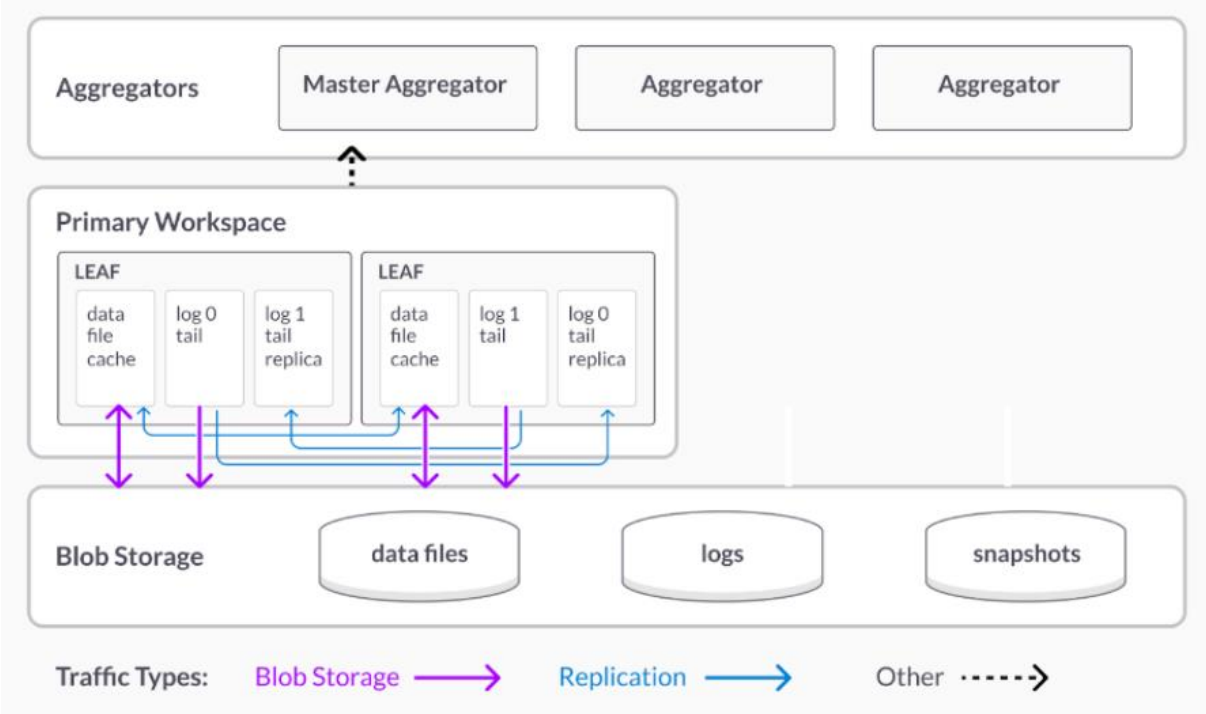


Separation of Storage and Compute

- Blob storage and local storage
  - Without blob storage, like a typical shared nothing architecture
  - Newly written data is persisted on the local storage moved to blob storage asynchronously
- Fast in-cluster replication and log pages replicated out of order
- Data for a partition of a columnstore stored as an LSM tree
  - Lower levels have persistent data stored on disk and logs and datafiles of the columnstore
  - Top levels have the in memory rowstore

Staging Data from Local to Remote Storage

- Transactions are committed to the tail of the log
- Hot data files are kept in a cache & cold data files are moved to blob storage
- Snapshots of rowstore data are taken only on master partitions
- New replicas take snapshots and logs they need from blob storage and replicate tail of the log



Capabilities of Separated Storage

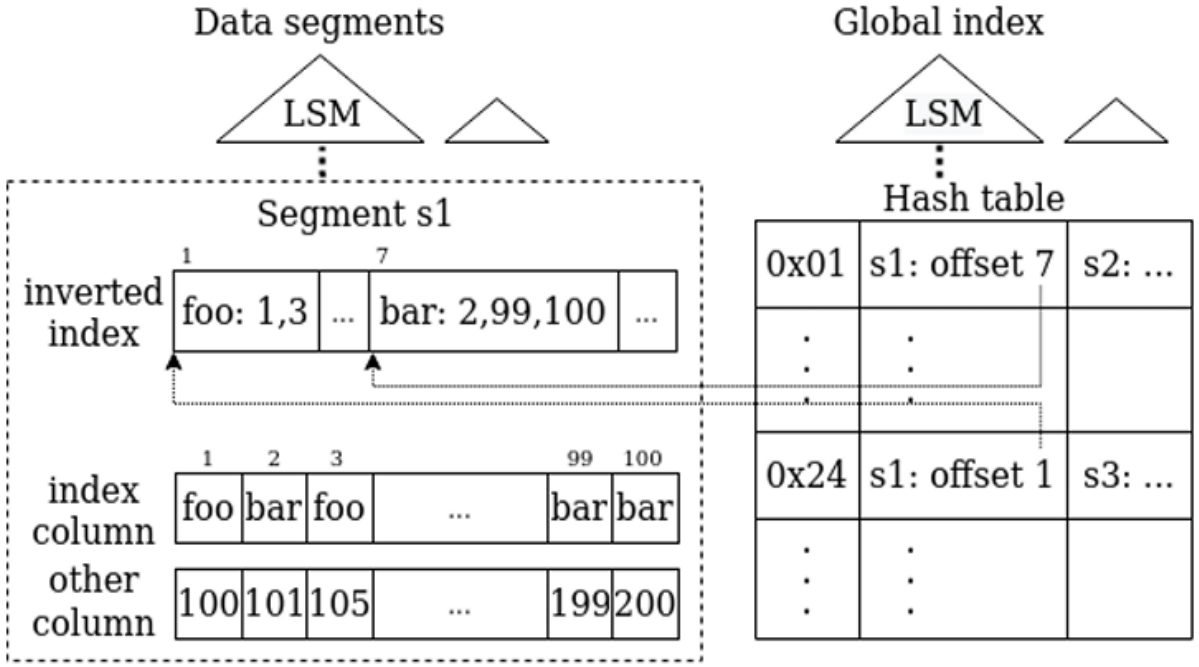
- Faster ephemeral SSDs for local storage
- Stores history/ Blob storage acts as backup
- Enables PITR to a previous state
- Supports creation of read only workspaces

Unified Table Storage

- Single table that internally makes use of rowstore and columnstore
- Works well for OLTP, OLAP and demanding HTAP transactions
- Columnstore data organized into:
  - LSM trees : avoids merge based reconciliation for reads, minimize disk access
  - Secondary indexes
- Extensive set of features supporting sort keys, secondary keys, shard keys, unique keys and row level locking

Secondary indexes

- Commonly used structures: per segment filtering index and external index structure
- Inverted index – maps indexed column values to a posting list
- Global index – maps values of indexed columns to the ids of segments
- Row locking for updating/deleting rows



Adaptive query execution

- Multiple ways to access data on the HTAP database
- Query execution engine combines and applies in optimal order
- S2DB adopts adaptive query engine for data access which performs:
  - Finding list of segments to read – Secondary indexes or min/max values from segment metadata
  - Filtering to find rows from segments – regular filtering, encoded filtering, group filtering, secondary index filtering
  - Selectively decodes and outputs rows
  - Select optimal strategy by costing each method on a small batch

Experimental Results

- CDW1, CDW2 – Two cloud data warehouses for comparison on TPC-C benchmarking with 1000 warehouses
- CDB – cloud operational database for TPC-H benchmarking
- Ran using S2DB's unified table storage: used indexes, sort key and shard key
- Comparison: TPC-H at scale factor of 1TB and TPC-C at 1000 warehouses
- S2DB scales better in both

Product	vCPU	Size (warehouses)	Throughput (tpmC)	Throughput (% of max)
CDB	32	1000	12,582	97.8%
S2DB	32	1000	12,556	97.7%
S2DB	256	10000	121,432	94.4%

Table 1: TPC-C results (higher is better, up to the limit of 12.86 tpmC/warehouse)

Product	Cluster price per hour	TPC-H geomean (sec)	TPC-H geomean (cents)	TPC-H throughput (QPS)
S2DB	$16.50	8.57 s	3.92 ¢	0.078
CDW1	$16.00	10.31 s	4.58 ¢	0.069
CDW2	$16.30	10.06 s	4.55 ¢	0.082
CDB	$13.92	Did not finish within 24 hours		

Table 2: Summary of TPC-H (1TB) results

- TPC-C specifies a maximum possible result of 12.86 tpmC per warehouse at 1000 warehouses
- Scales linearly at 10000 warehouses
- S2DB achieves competitive performance with leading databases
- S2DB, CDW1 and CDW2 all have competitive performance on TPC-H

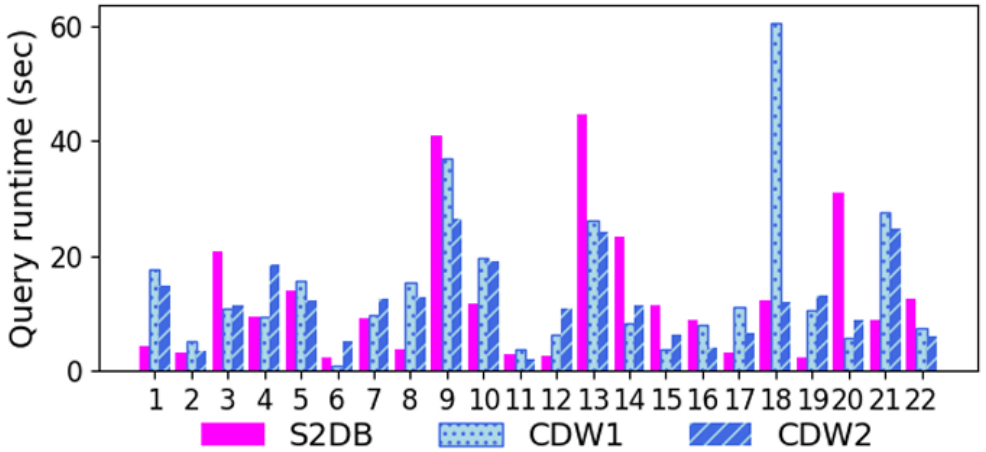


Figure 4: TPC-H 1TB query runtimes (lower is better)

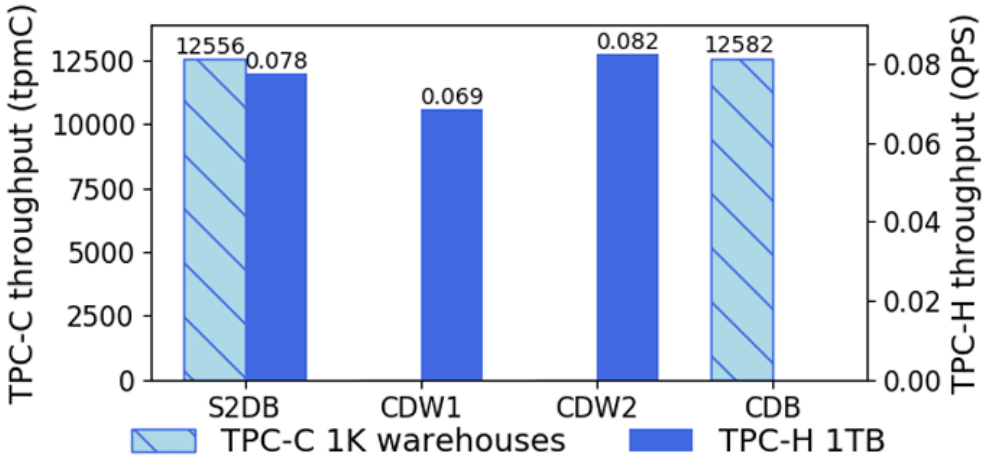


Figure 5: Summary of TPC-C and TPC-H throughputs (higher is better)

Success Story

SingleStoreDB is used by a variety of companies across a wide range of industries, including:

- Technology: Dell, HP, Siemens, GE, NVIDIA, Airbnb, Uber
- Financial services: Goldman Sachs, Morgan Stanley, Citigroup, Bank of America
- Retail: Walmart, Target, Home Depot, Lowe's, Kroger
- Healthcare: Kaiser Permanente, UnitedHealth Group, CVS Health, Walgreens Boots Alliance
- Media and entertainment: Disney, WarnerMedia, Netflix, Spotify
- Telecommunications: AT&T, Verizon, T-Mobile, Sprint

Thoughts and Questions?