Studying state-of-the-art HTAP systems

- Satya Sai Bharath Vemula
- Rwitam Bandyopadhyay
- Shubham Pandey

Table of Contents

PolarDB

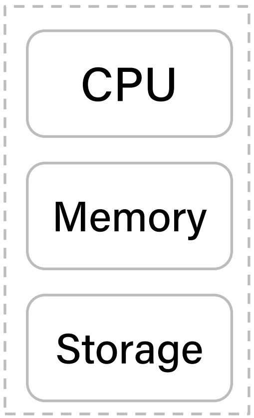
Disagg1

Presented by Shubham Pandey

PURDUE UNIVERSITY

PolarDB

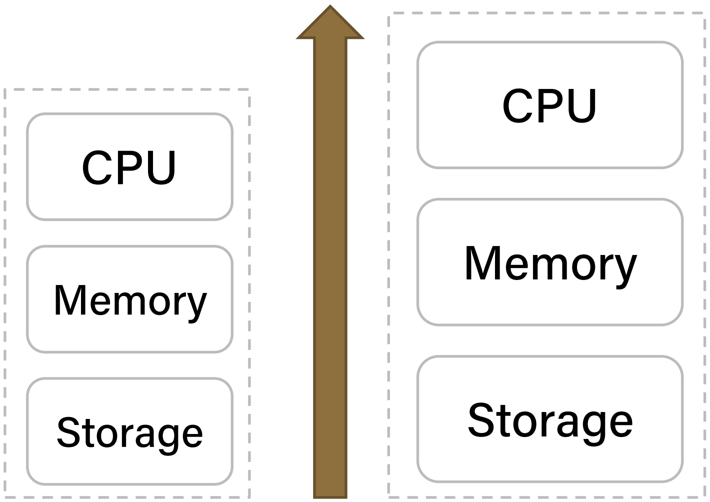
Introduction



Source: https://www.alibabacloud.com/blog/analyze-the-technical-essentials-of-polardb-at-the-architecture-level_599347

PolarDB

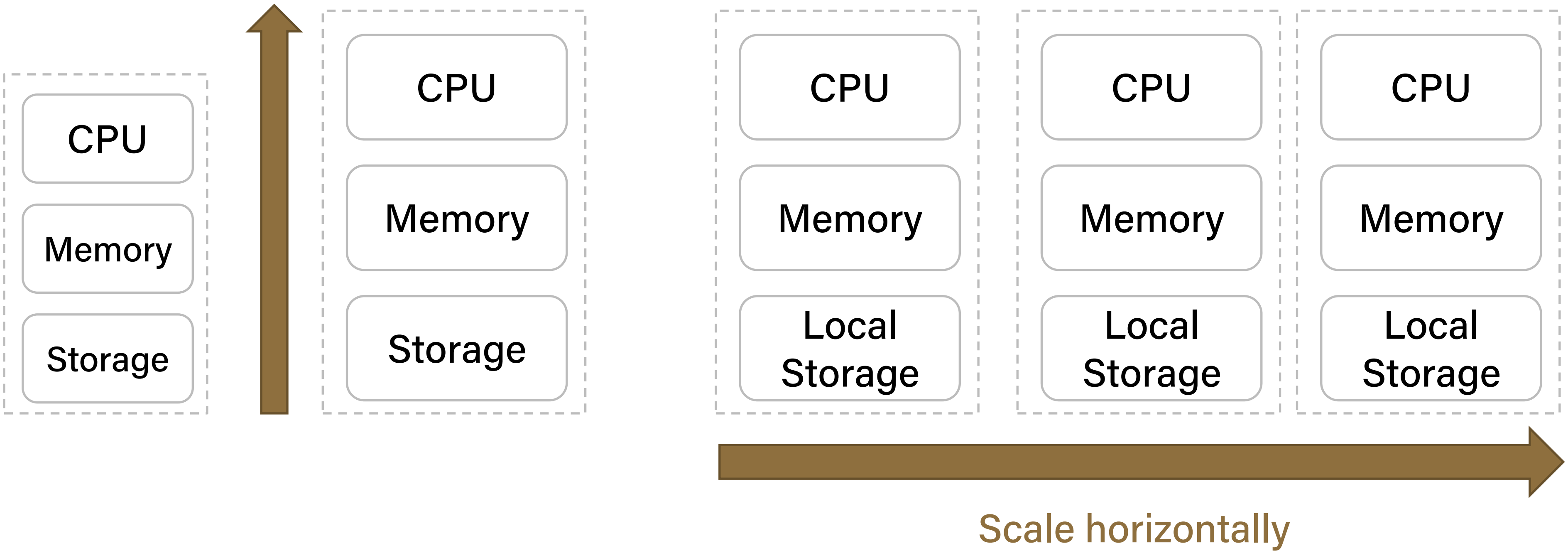
Introduction



Scale vertically

Source: https://www.alibabacloud.com/blog/analyze-the-technical-essentials-of-polardb-at-the-architecture-level_599347

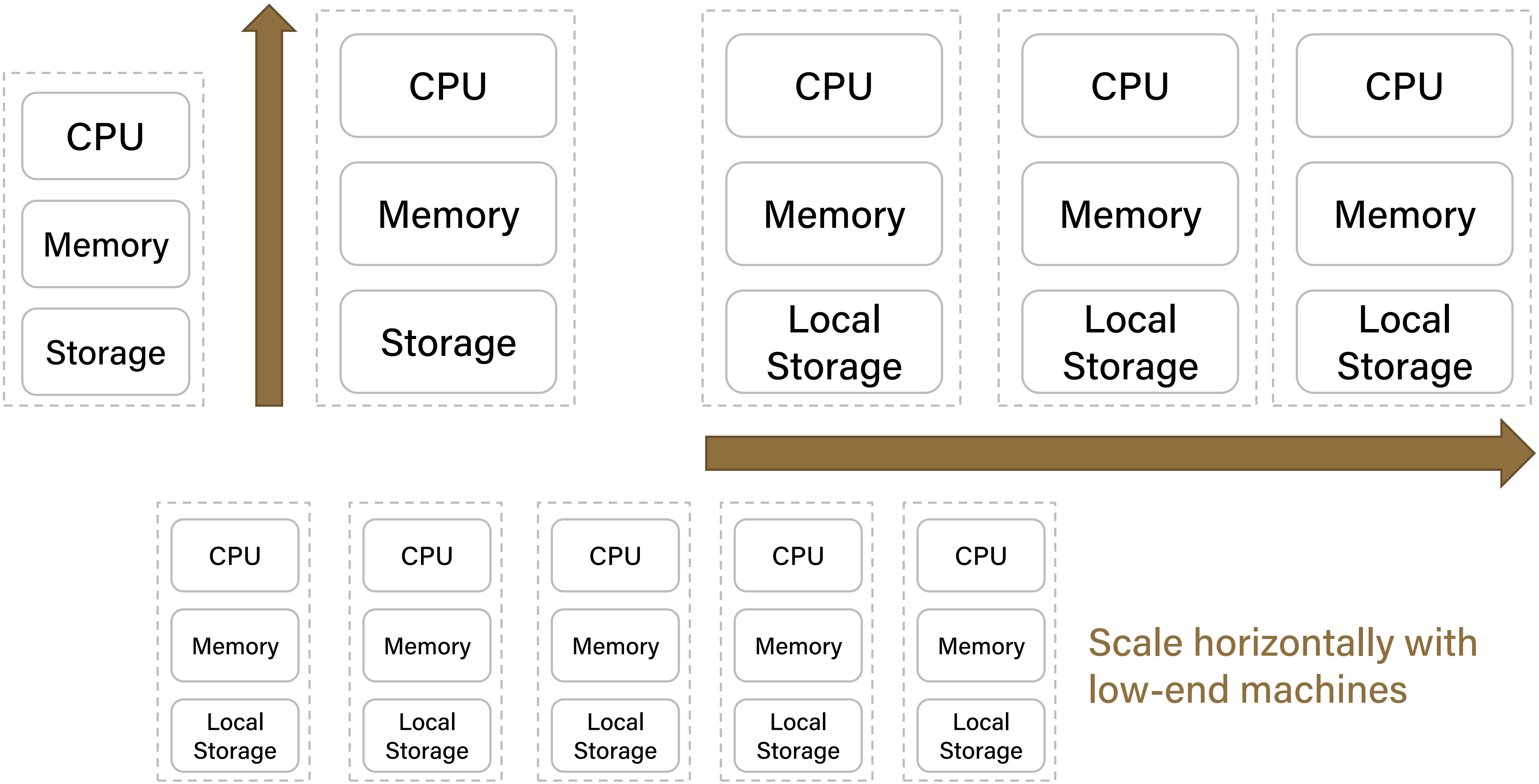
Introduction



Source: https://www.alibabacloud.com/blog/analyze-the-technical-essentials-of-polardb-at-the-architecture-level_599347

PolarDB

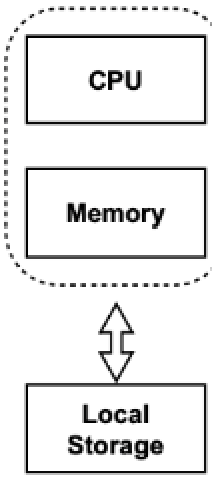
Introduction



Scale horizontally with low-end machines

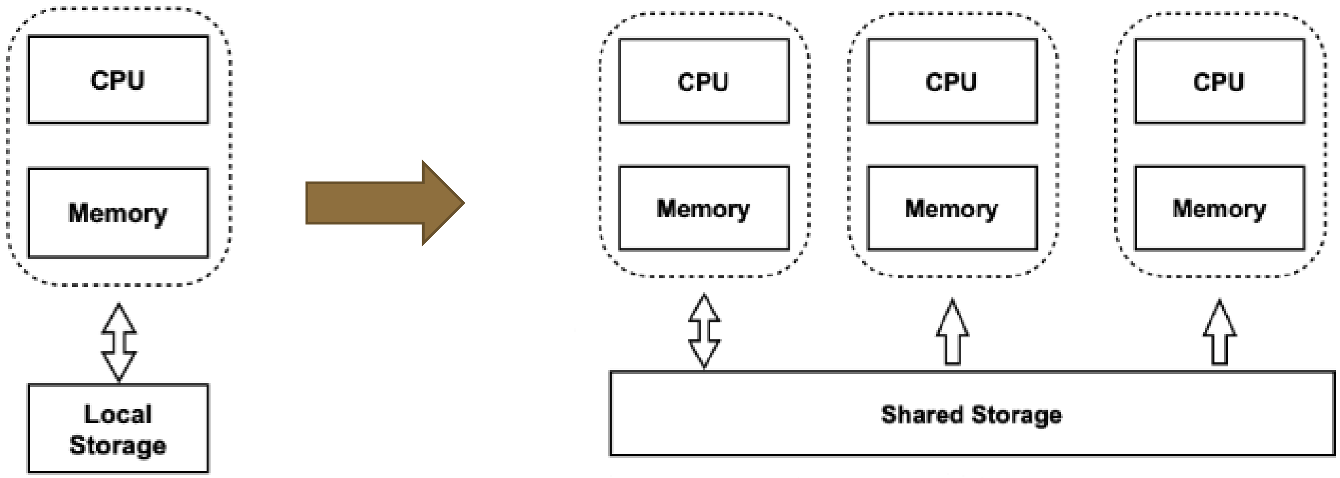
Source: https://www.alibabacloud.com/blog/analyze-the-technical-essentials-of-polardb-at-the-architecture-level_599347

Introduction



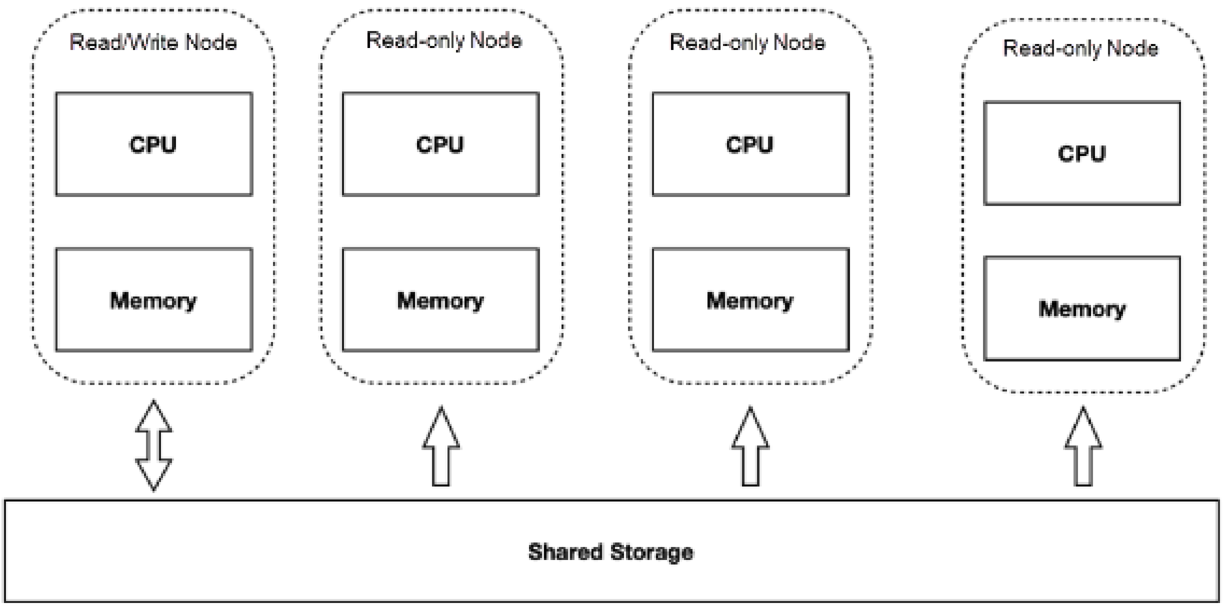
Source: https://www.alibabacloud.com/blog/analyze-the-technical-essentials-of-polardb-at-the-architecture-level_599347

Introduction



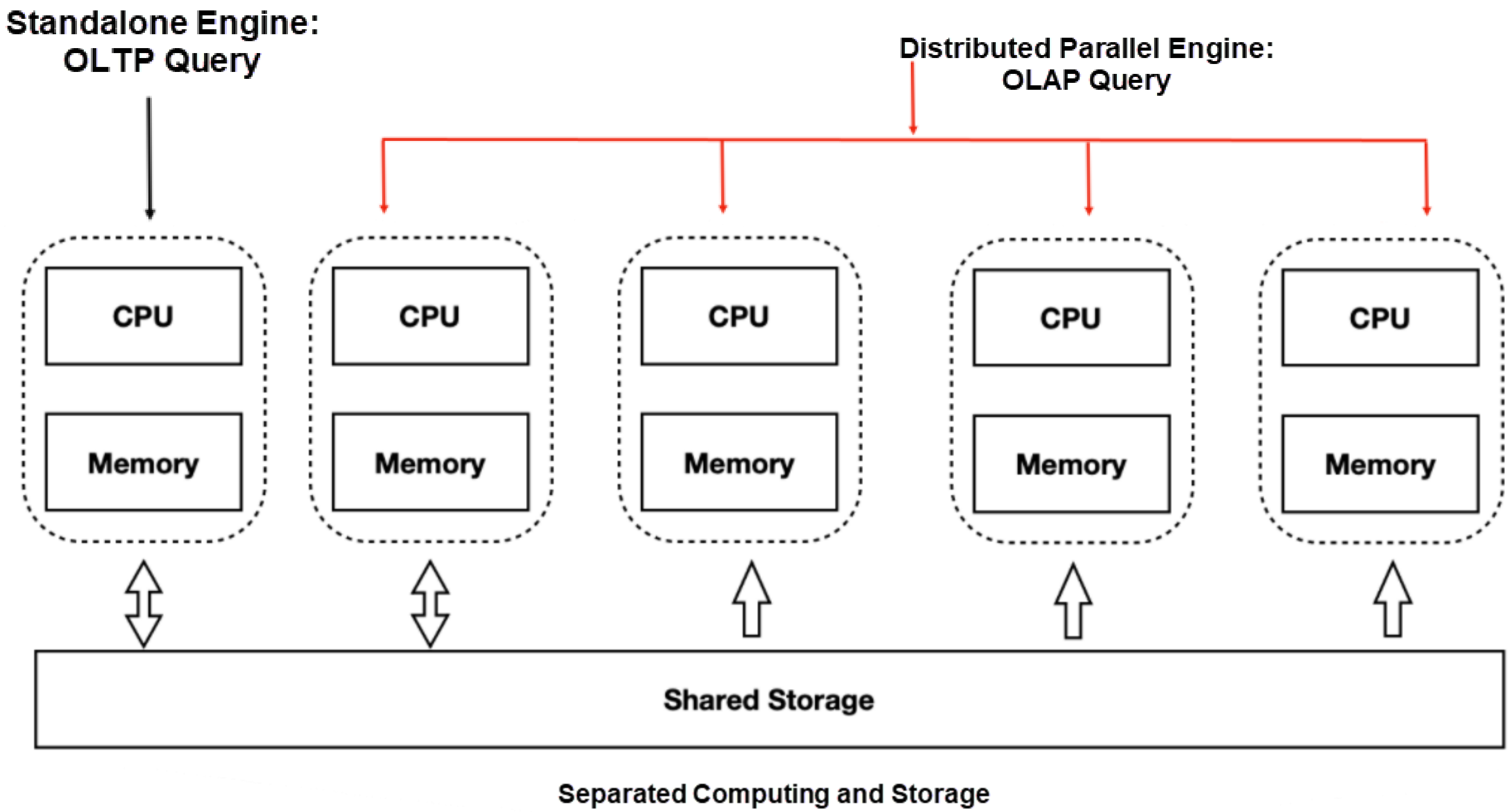
Source: https://www.alibabacloud.com/blog/analyze-the-technical-essentials-of-polardb-at-the-architecture-level_599347

Shared Storage Architecture



Source: https://www.alibabacloud.com/blog/analyze-the-technical-essentials-of-polardb-at-the-architecture-level_599347

HTAP



Source: https://www.alibabacloud.com/blog/analyze-the-technical-essentials-of-polardb-at-the-architecture-level_599347

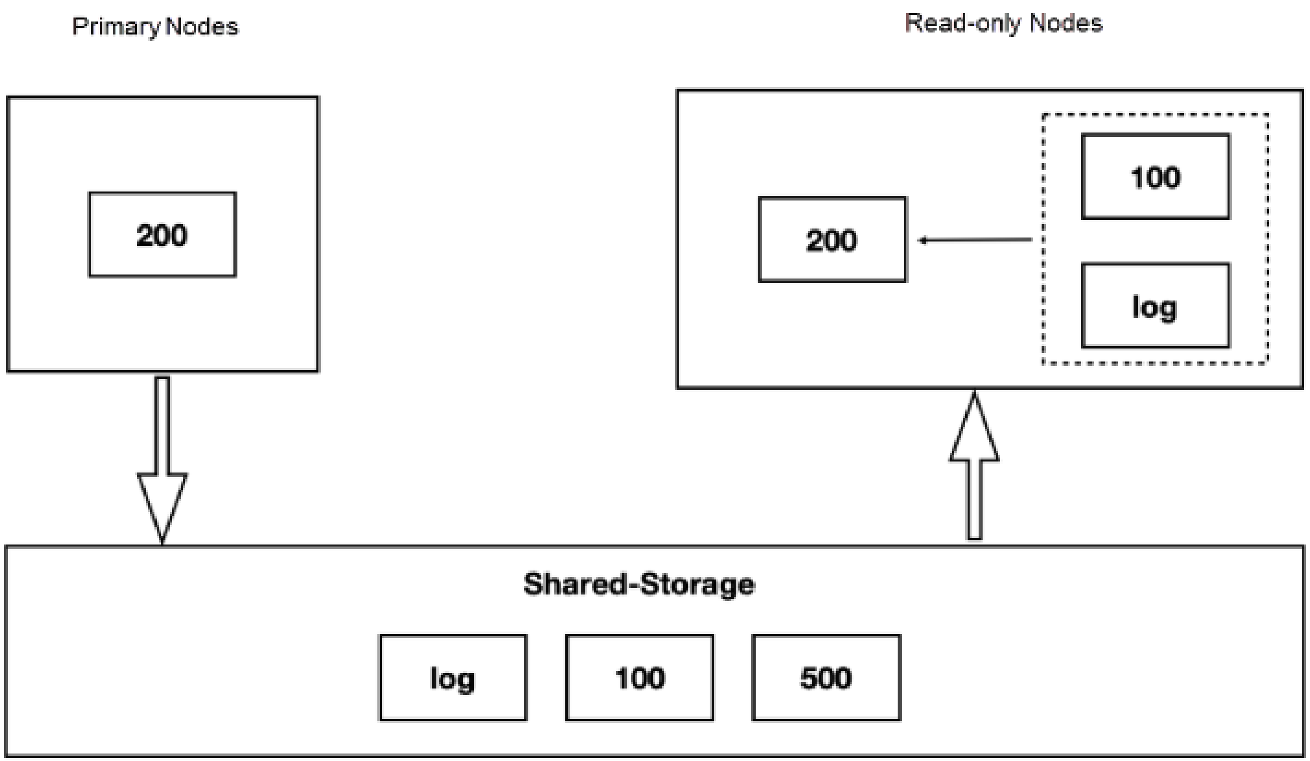
Challenges Posed by Shared Storage

- Data Consistency
- Read/Write Splitting
- High Availability

Source: https://www.alibabacloud.com/blog/analyze-the-technical-essentials-of-polardb-at-the-architecture-level_599347

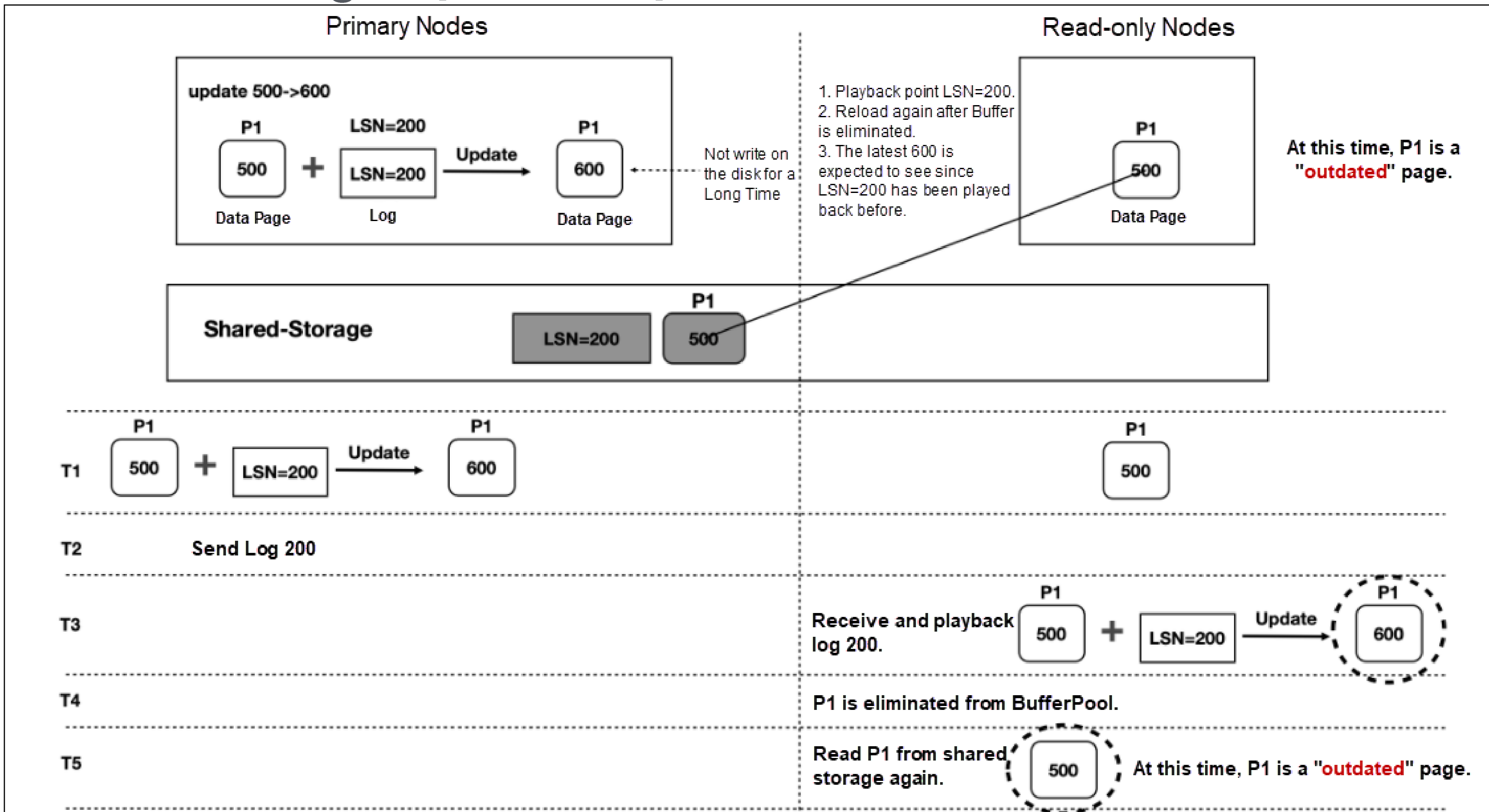
PolarDB

Data Consistency



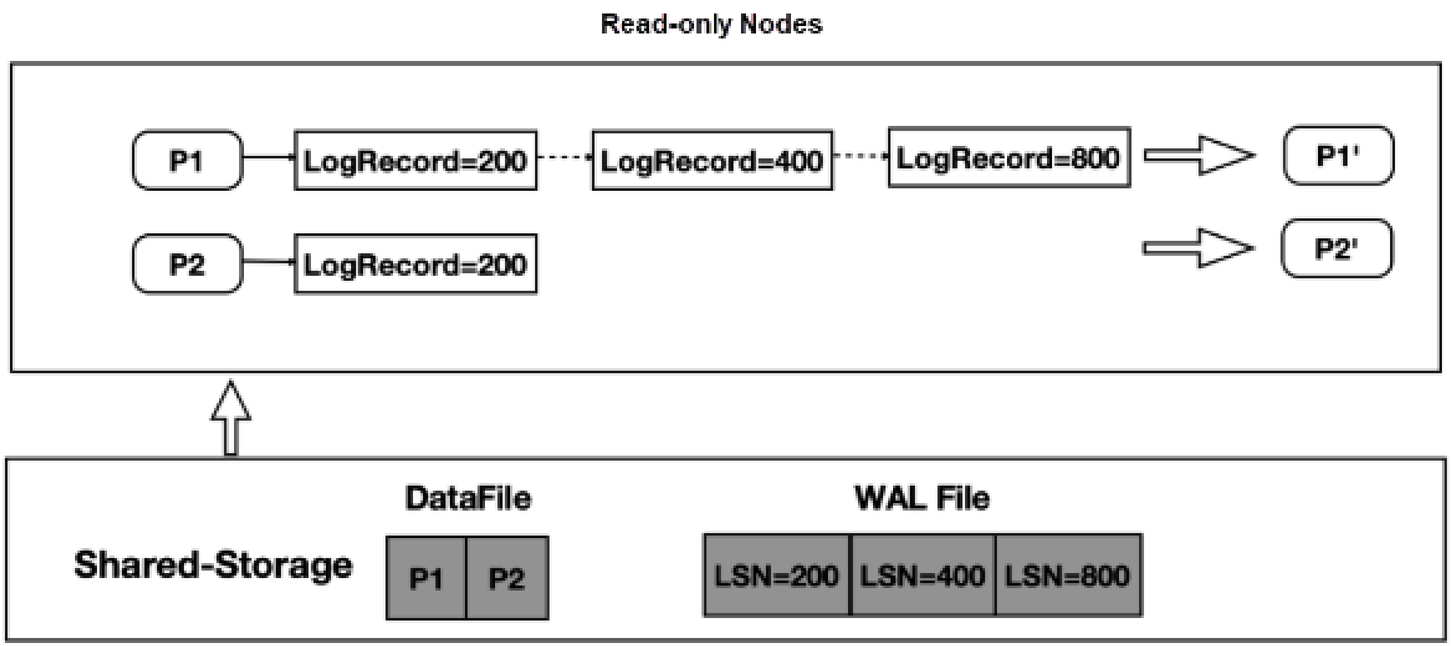
Source: https://www.alibabacloud.com/blog/analyze-the-technical-essentials-of-polardb-at-the-architecture-level_599347

Outdated Pages (Problem)



Source: https://www.alibabacloud.com/blog/analyze-the-technical-essentials-of-polardb-at-the-architecture-level_599347

Outdated Pages (Solution)



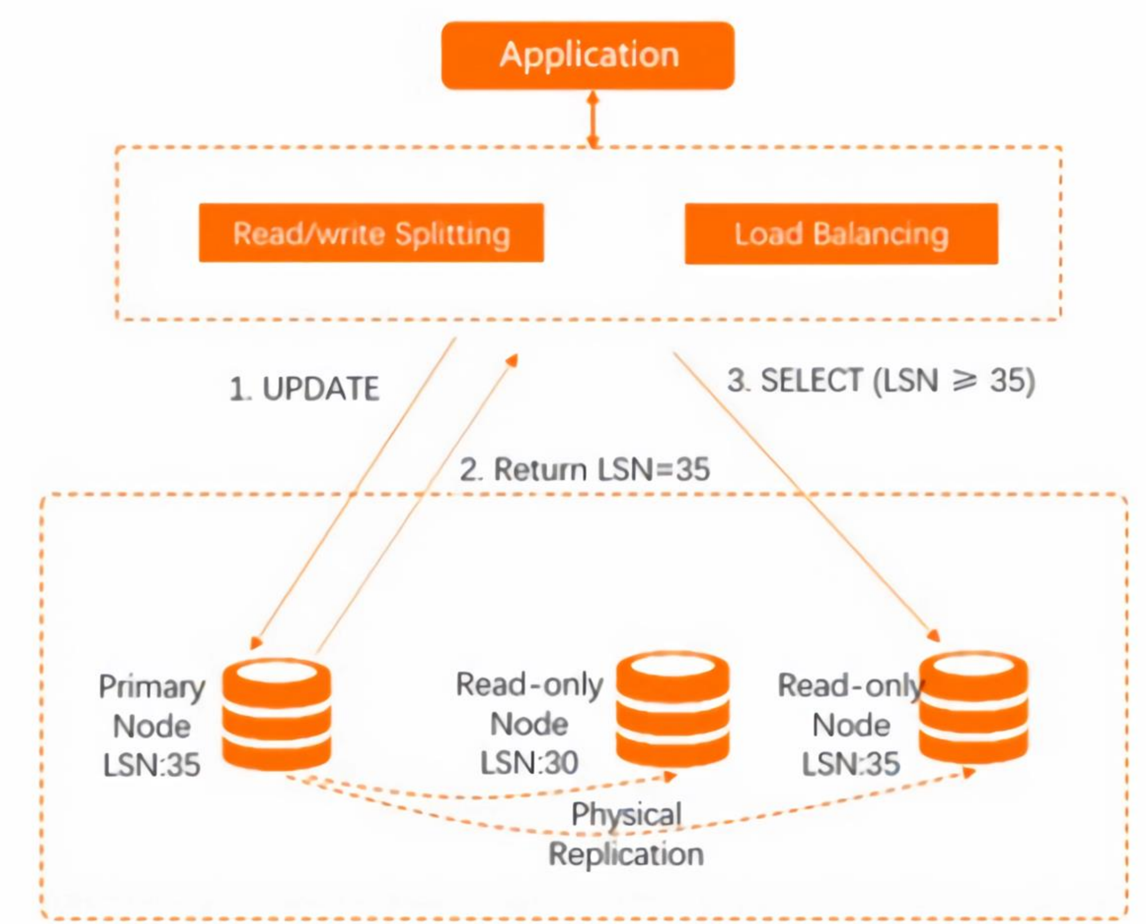
Source: https://www.alibabacloud.com/blog/analyze-the-technical-essentials-of-polardb-at-the-architecture-level_599347

Session Read Consistency (Problem)

```
connection query
{
    UPDATE user SET name = "Jimmy" WHERE id=1;
    COMMIT
    SELECT name FROM user WHERE id=1;
}
```

Source: PolarDB Series (5): Session Read Consistency - Alibaba Cloud Community

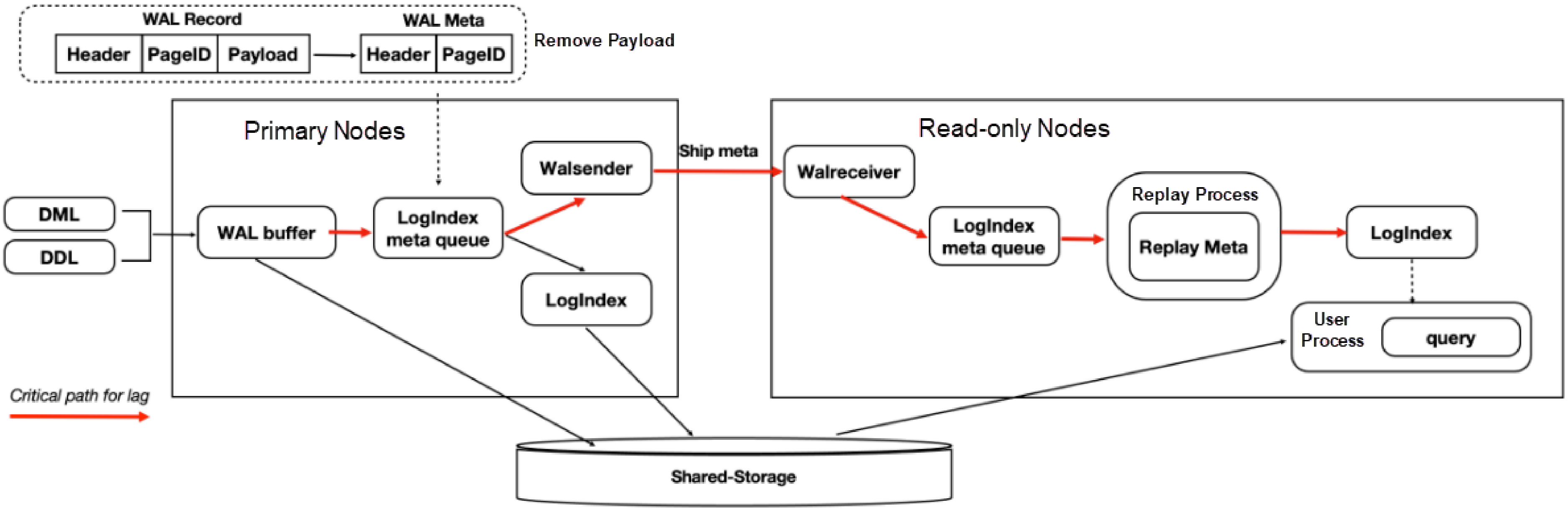
Session Read Consistency (Solution)



Source: [PolarDB Series (5): Session Read Consistency - Alibaba Cloud Community]

Low Latency Replication (1)

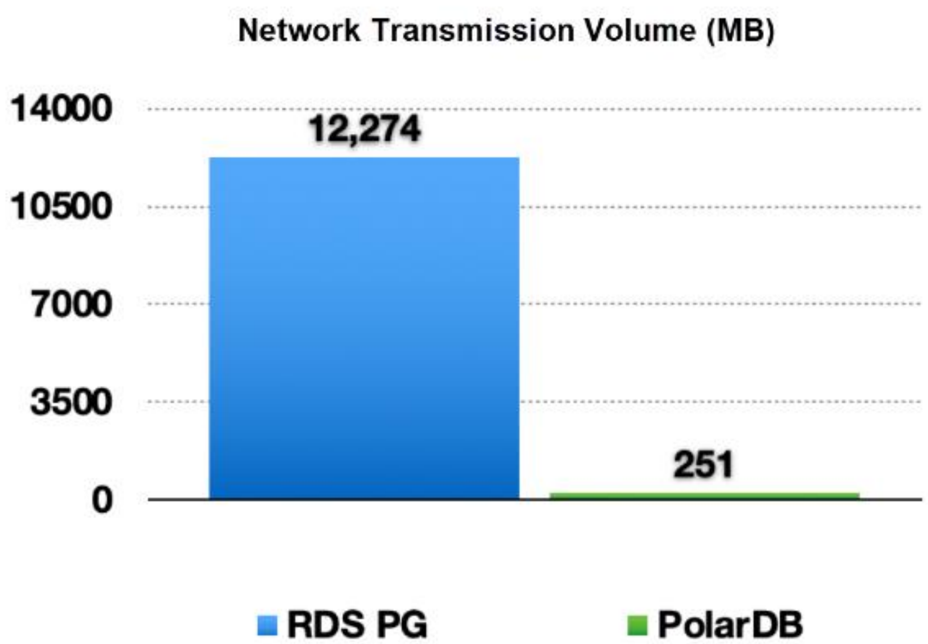
Copy Meta Only



Source: https://www.alibabacloud.com/blog/analyze-the-technical-essentials-of-polardb-at-the-architecture-level_599347

Low Latency Replication (1)

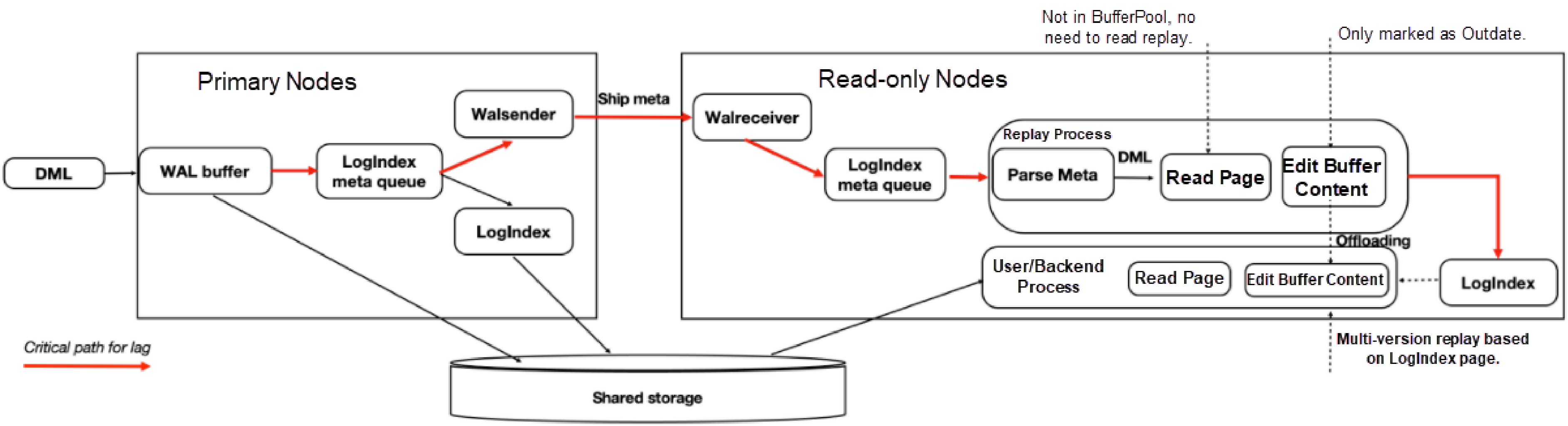
Copy Meta Only



Source: https://www.alibabacloud.com/blog/analyze-the-technical-essentials-of-polardb-at-the-architecture-level_599347

Low Latency Replication (2)

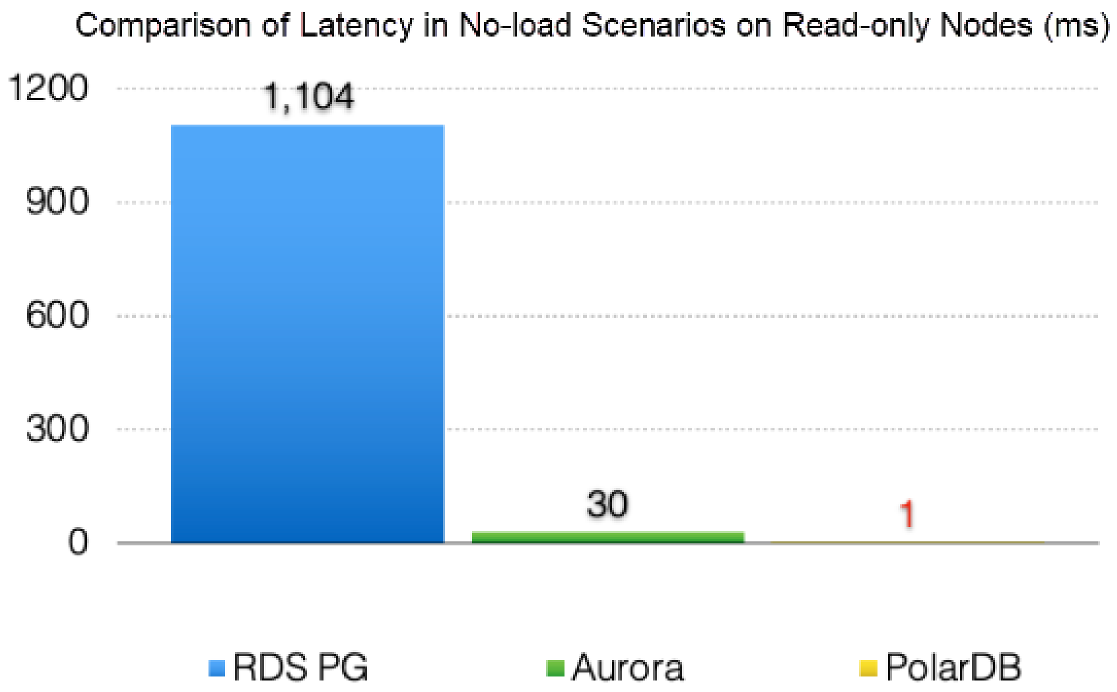
Page Replay Optimization



Source: https://www.alibabacloud.com/blog/analyze-the-technical-essentials-of-polardb-at-the-architecture-level_599347

Low Latency Replication (2)

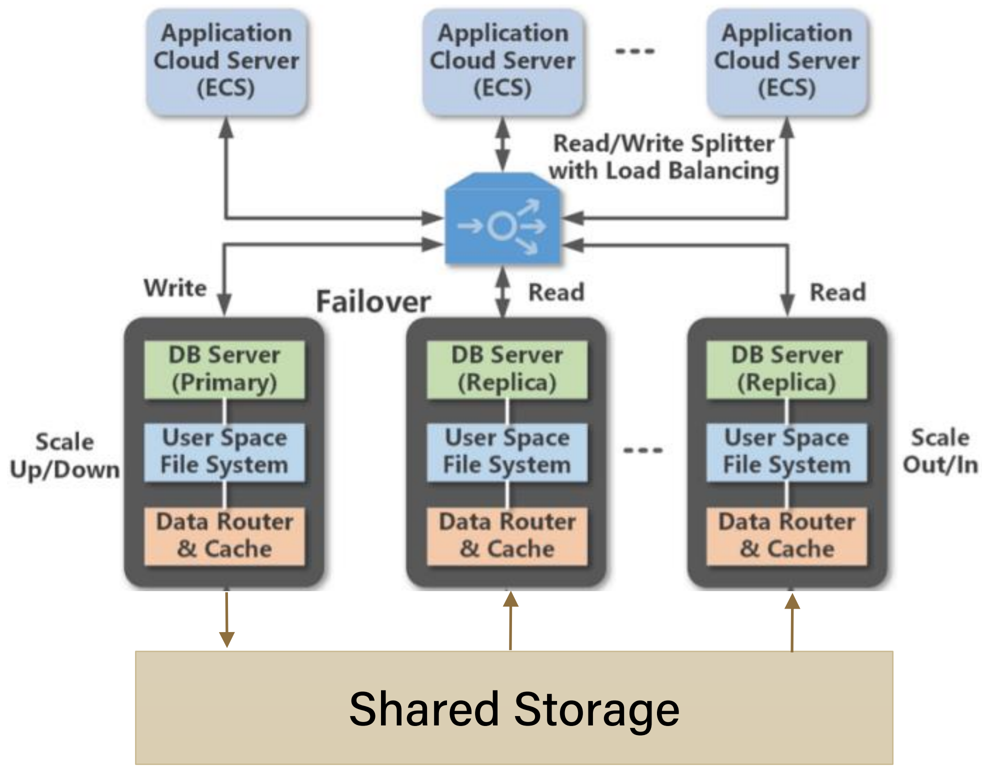
Page Replay Optimization



Source: https://www.alibabacloud.com/blog/analyze-the-technical-essentials-of-polardb-at-the-architecture-level_599347

Fault Tolerance

Active-Active-Failover

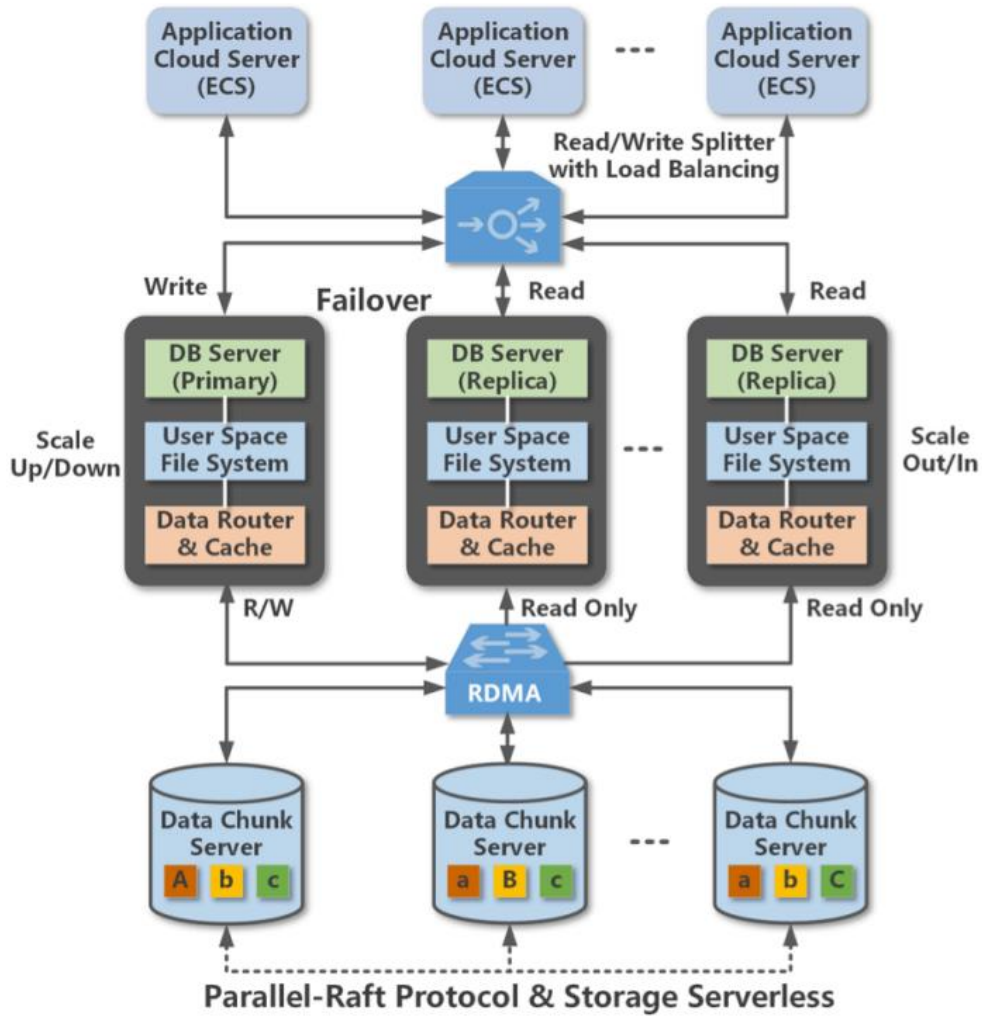


Source: https://www.alibabacloud.com/blog/analyze-the-technical-essentials-of-polardb-at-the-architecture-level_599347

PolarDB

Shared Storage

- Chunk Servers
- Proxy
- RDMA
- Parallel-Raft



Source: https://www.alibabacloud.com/blog/analyze-the-technical-essentials-of-polardb-at-the-architecture-level_599347

SingleStore

Disagg1

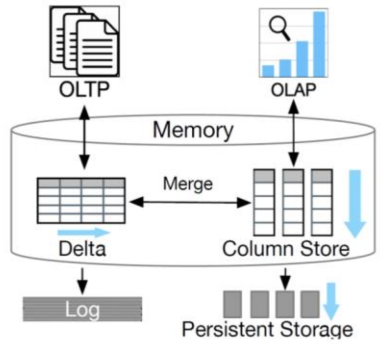
Presented by Satya Sai Bharath Vemula

PURDUE UNIVERSITY

SingleStore

Introduction / System Features

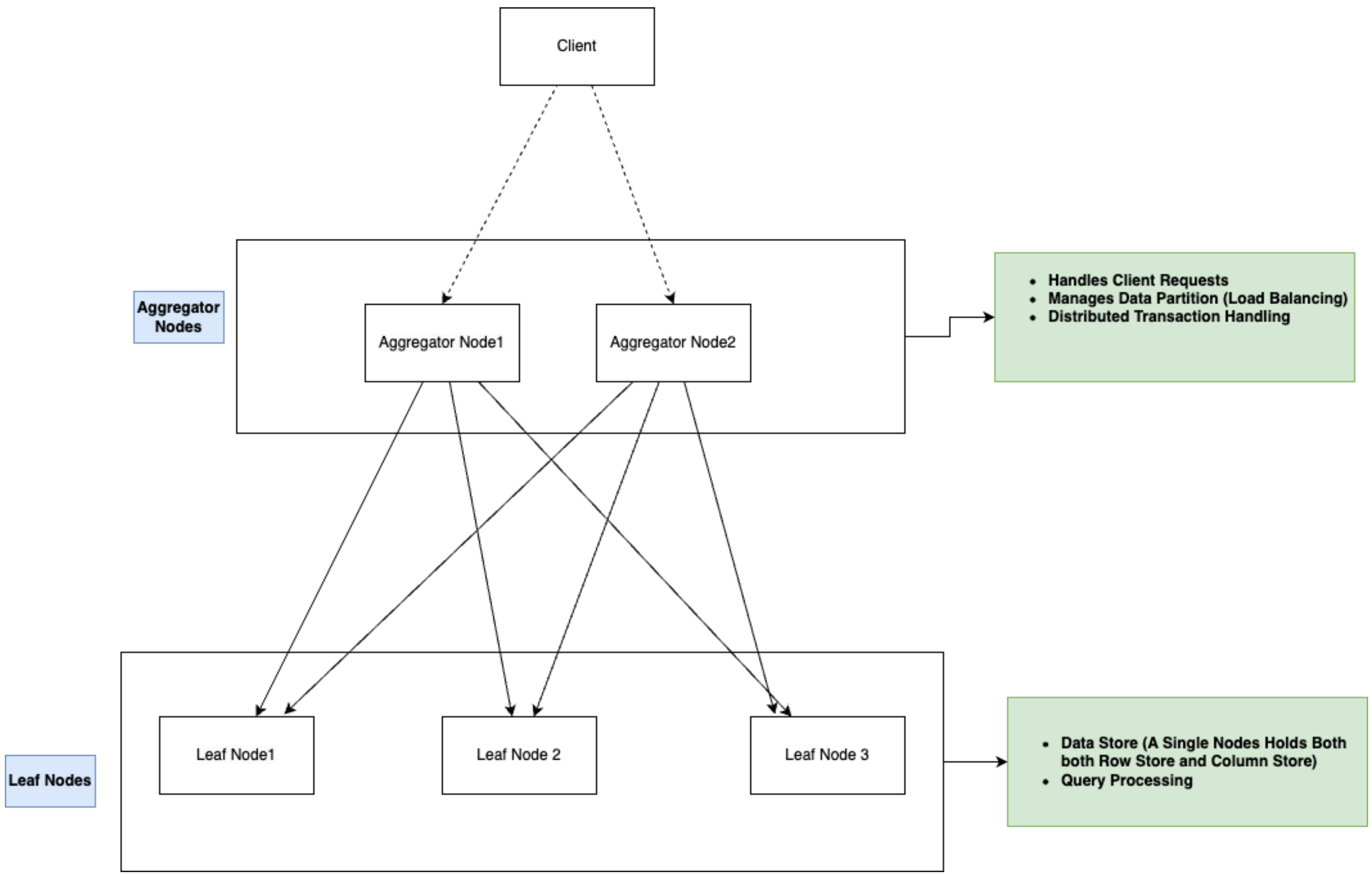
- A Cloud native HTAP Database, Upgraded version of MemSQL.
- Column Store is the first-class citizen (i.e., data is persisted in columnar format), Row store is stored in memory
- Highly-scalable Distributed database
- SQL Compatible



Primary Column Store + Delta Row Store

Source: [HTAP Databases: What is New and What is Next | Proceedings of the 2022 International Conference on Management of Data (acm.org)]

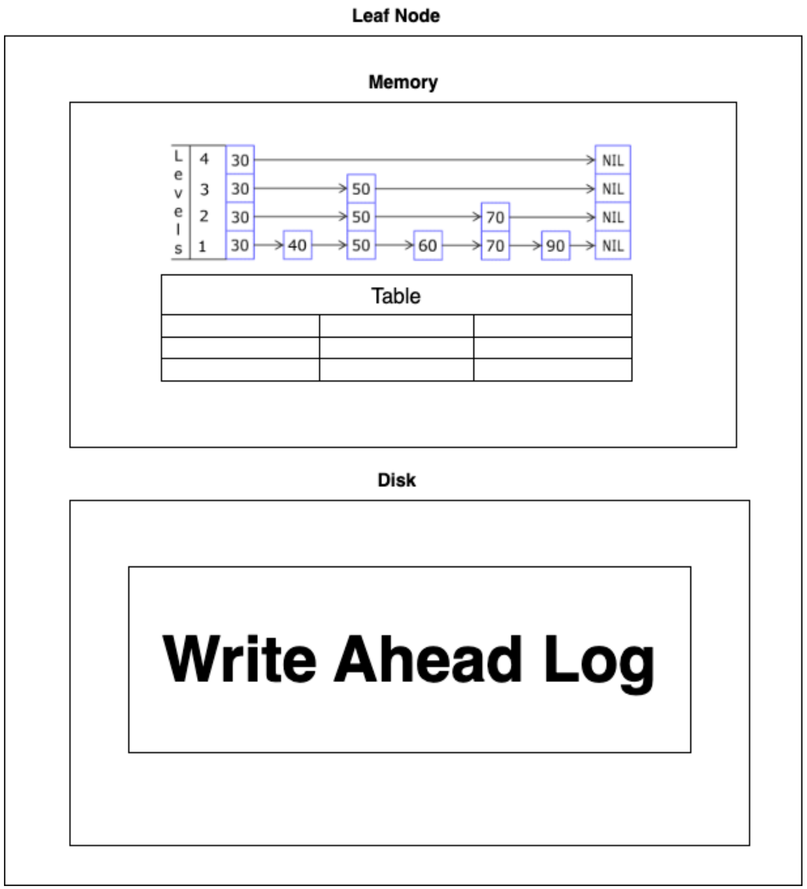
Architecture



SingleStore

Row Store

- OLTP queries are honored using the in-memory row store
- Row store in the memory is built as a skip list index Table (ordered on key) where the leaf nodes contain the row tuple
- Write ahead log ensures changes are persisted to disk
- Since Memory if only a fraction of disk space, Only *Hot data* is placed inside the in-memory Row store



Source: https://saince.io/2022/10/19/singlestore-the-one-ring-to-rule-them-all/

Column Store & Data Sync

columnstore data files. To explain how the integrated durability

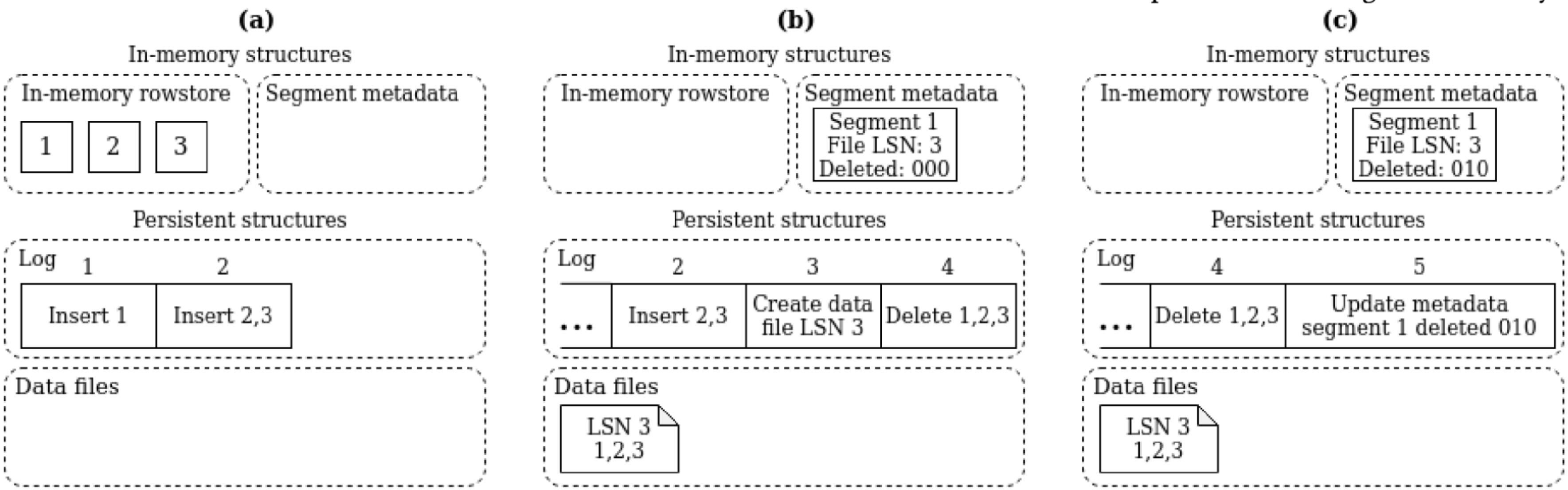
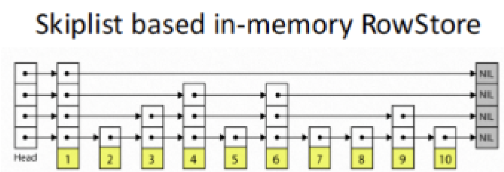
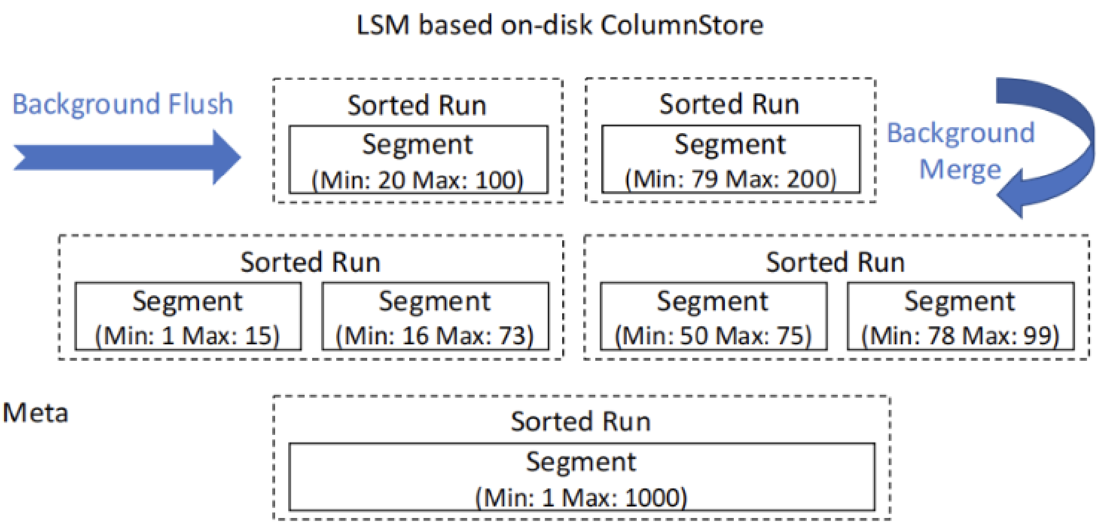


Figure 1: Example of a sequence of writes in S2, showing the state of the in memory and persisted structures in each step. (a) Inserting rows 1,2,3 in two transactions (b) Converting in-memory rows 1,2,3 to segment 1 (c) Deleting row 2 from segment 1.

Column Store Merge

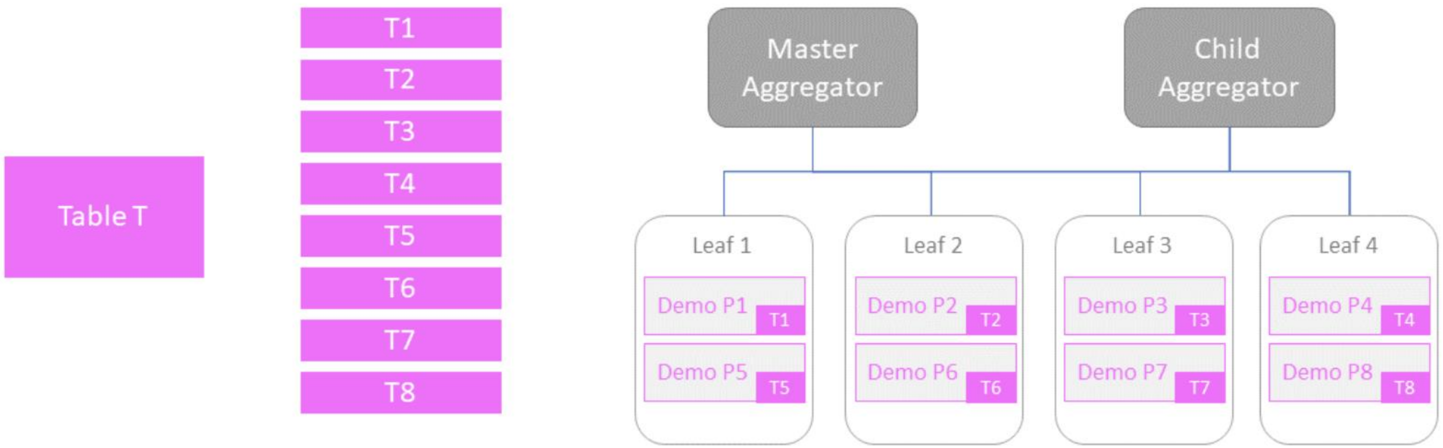


- Memory-Optimized
- MVCC
- Row-Level Locking
- Redo Log Backed
- For User Table Delta & ColumnStore Meta



Source: https://saince.io/2022/10/19/singlestore-the-one-ring-to-rule-them-all/

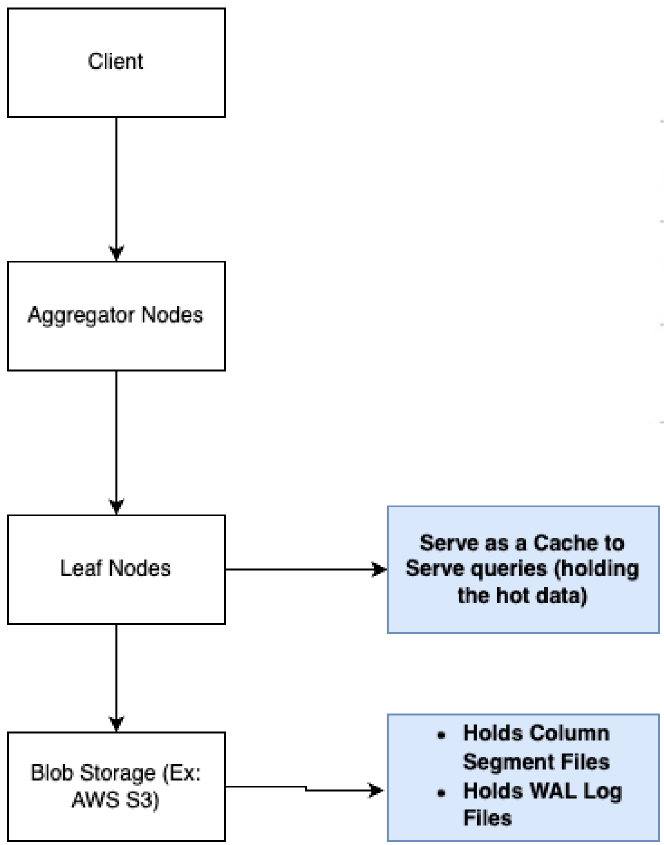
Scalability

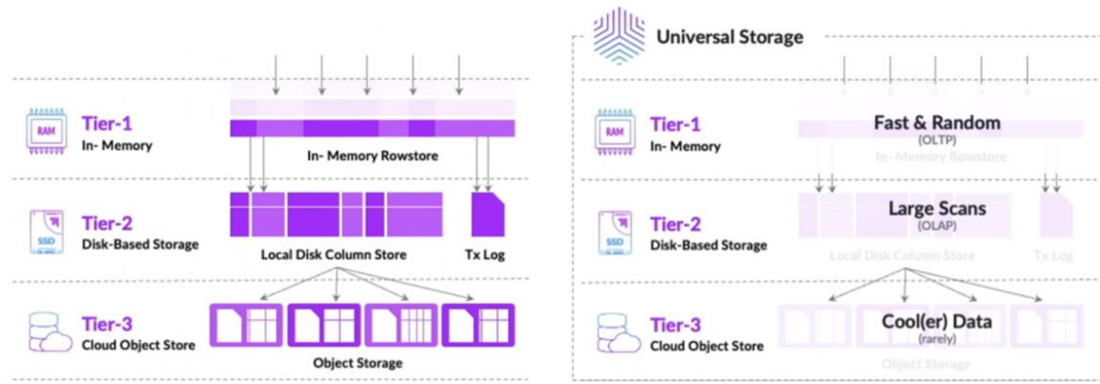


Source: https://saince.io/2022/10/19/singlestore-the-one-ring-to-rule-them-all/

SingleStore

Bottomless Architecture / 3 Tier Architecture

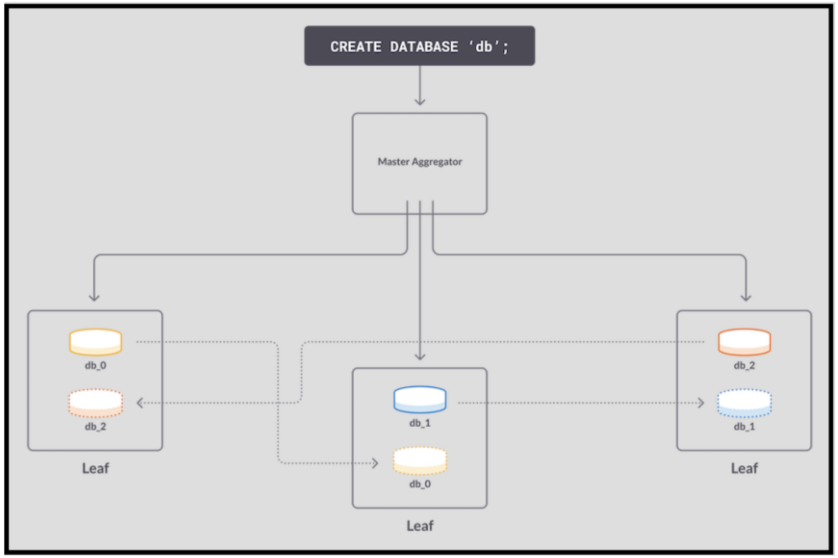




Source: https://saince.io/2022/10/19/singlestore-the-one-ring-to-rule-them-all/

Availability

- High Availability is enabled by Replication, with two node configuration
  - Primary
  - Replica
- Number of Replicas for each partition can be configurable
- Replication happens in a synchronous way.
- Replicas are by default only for Durability of data and are not responsible for handling client requests



Source: https://saince.io/2022/10/19/singlestore-the-one-ring-to-rule-them-all/

Availability (Read Replica)

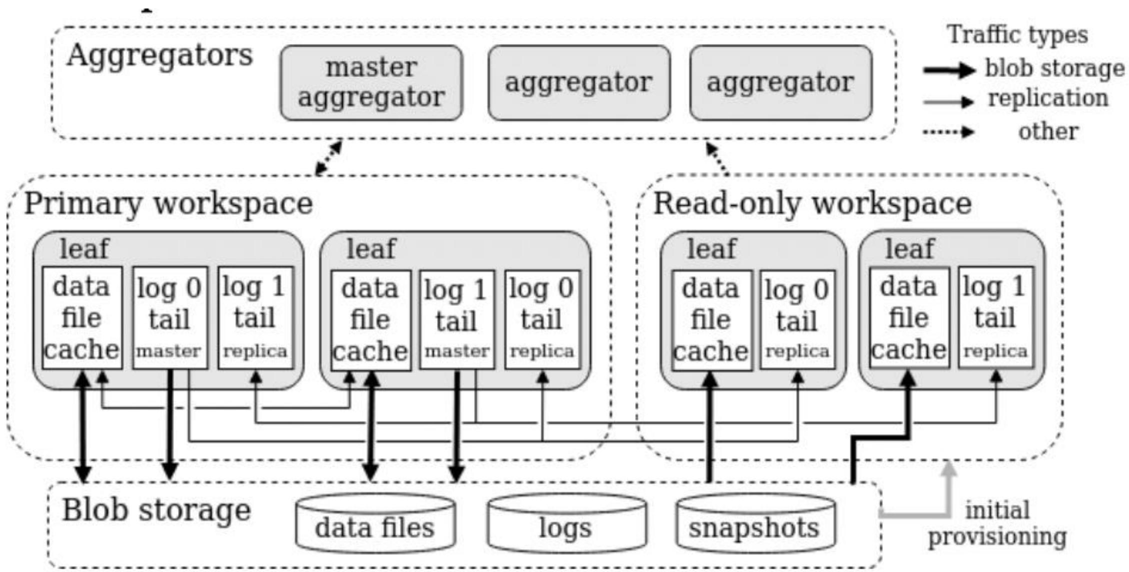
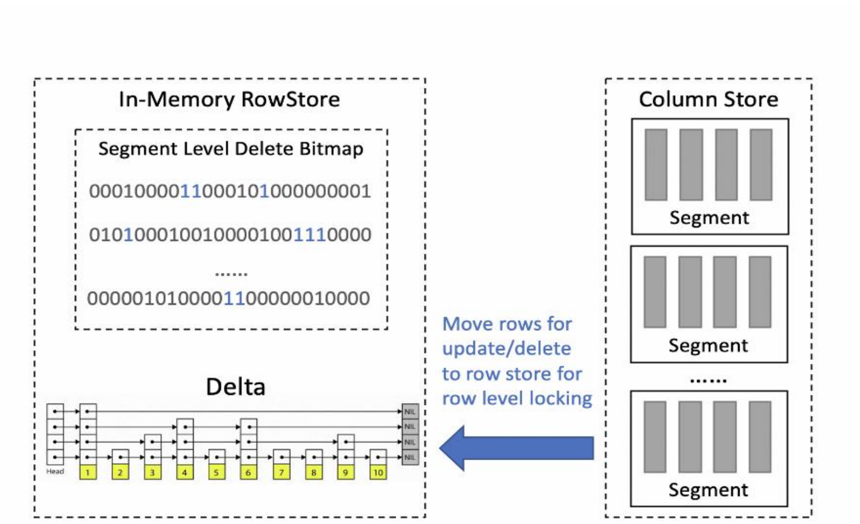
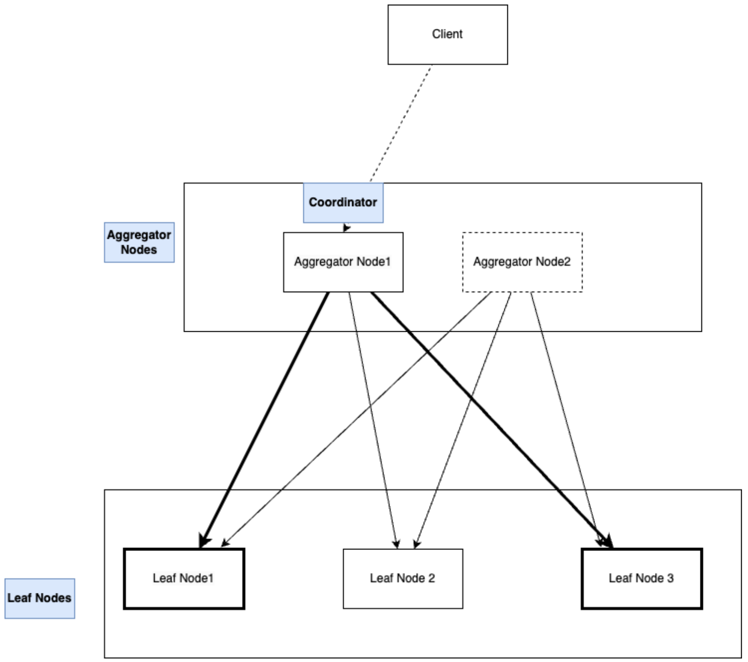


Figure 2: Cluster architecture with separated storage. The left side shows how the master workspace uploads data files, logs, and snapshots to blob storage asynchronously, while using replication to ensure durability of the log tails. The right side shows a read-only workspace provisioned from blob storage

Source: https://images.go.singlestore.com/Web/MemSQL/%7Bd1e77ba1-0e3e-4194-8438-2e289cb2b87c%7D_research-paper-cloud-native-transactions-and-analytics-in-singlestore.pdf

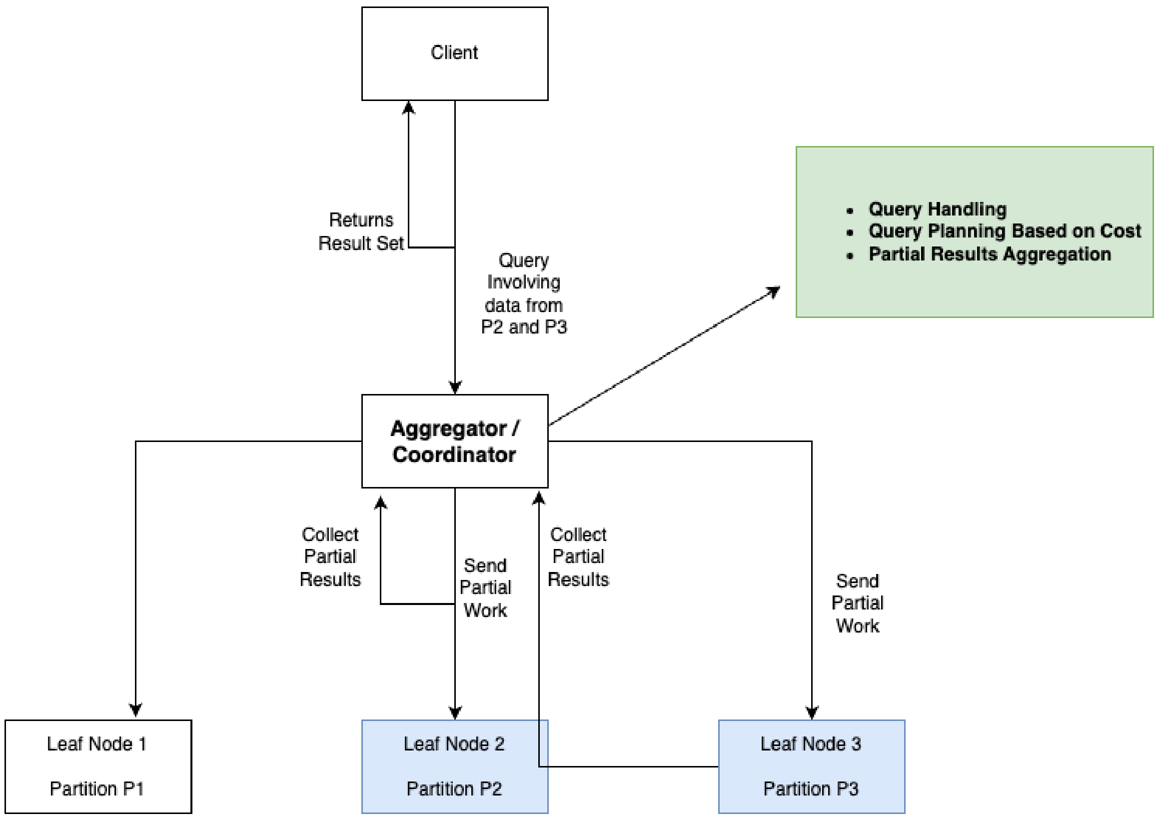
OLTP Query Processing





Source: https://images.go.singlestore.com/Web/MemSQL/%7Bd1e77ba1-0e3e-4194-8438-2e289cb2b87c%7D_research-paper-cloud-native-transactions-and-analytics-in-singlestore.pdf

OLAP Query Processing



Greenplum DB

Disagg1

Presented by Rwitam Bandyopadhyay

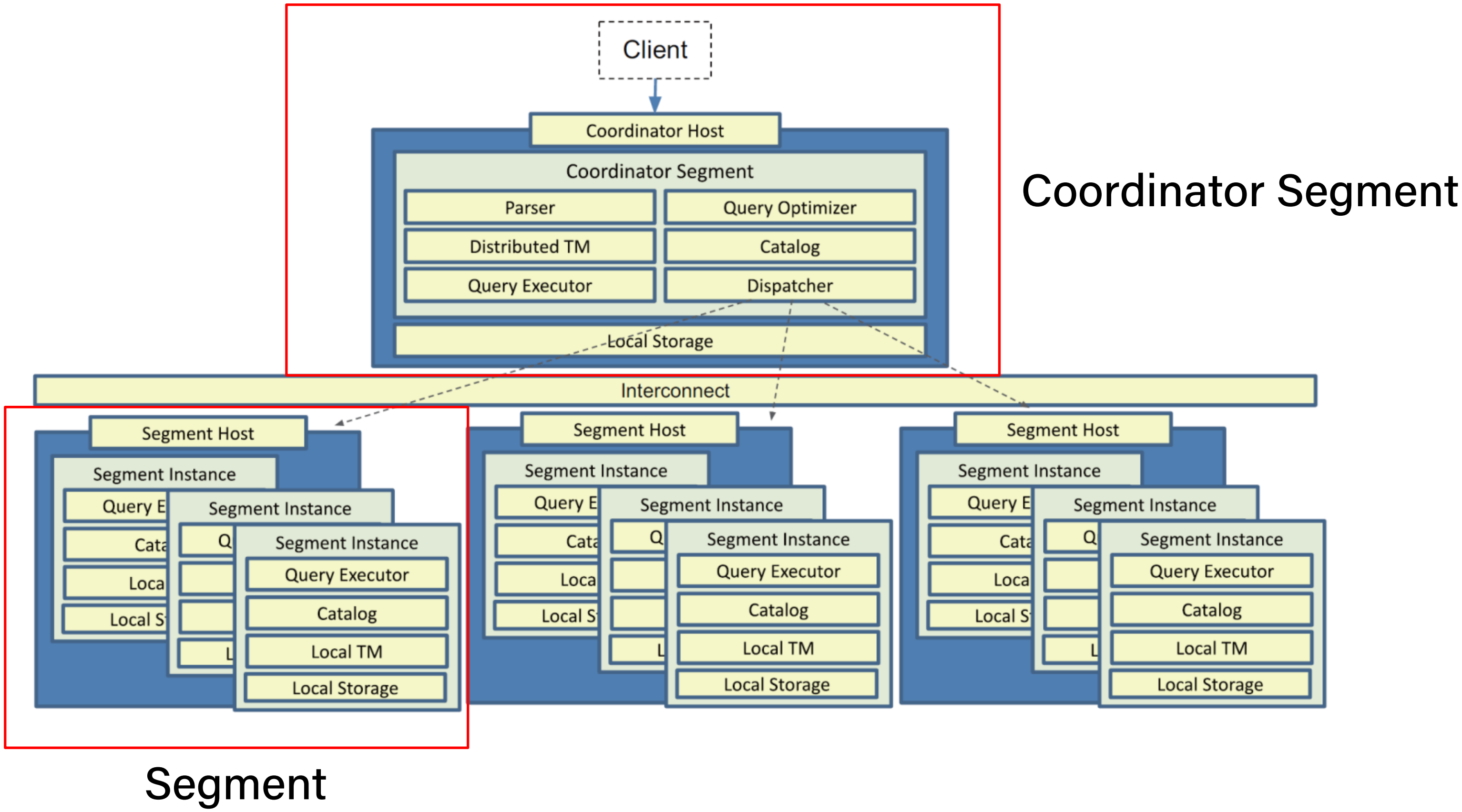
Background and History

- Greenplum is an established large-scale data-warehouse system.
- Designed with OLAP Queries as the first-class citizen.
- There is always a performance tradeoff between OLAP and OLTP.
- Greenplum Version 6 (introduced in 2019) featured massive gains in OLTP Performance. Below shows a community conducted comparison of INSERT performances.
- How an OLAP system was transformed into an HTAP system?

MySQL 5.6	Greenplum 6.2.1
1926/second	2252/second

Greenplum Architecture

It uses Massively Parallel Processing (MPP) Architecture

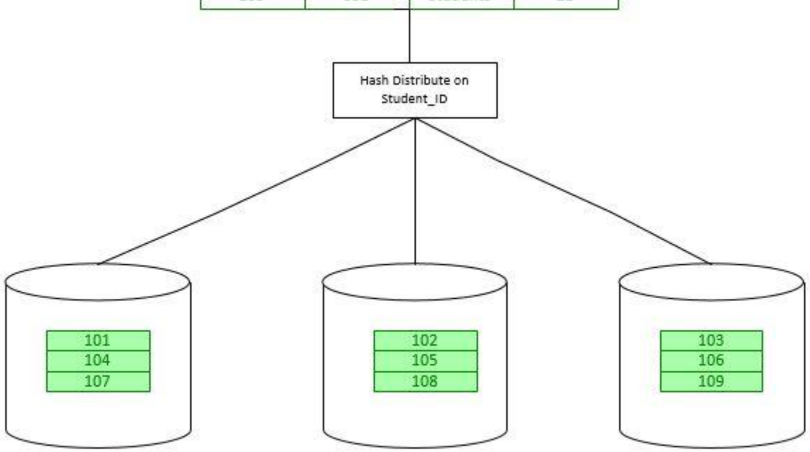


Source: Lyu, Zhenghua et al. Greenplum: A Hybrid Database for Transactional and Analytical Workloads, SIGMOD '21: Proceedings of the 2021 International Conference on Management of Data.

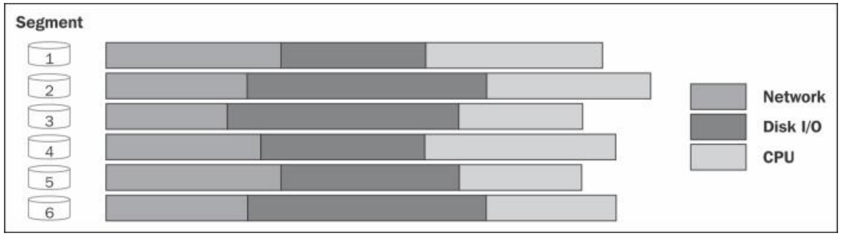
Greenplum Architecture (Continued)

Data Distribution

Student_id	Sub_code	Name	Marks
101	S01	Student1	50
102	S02	Student2	60
103	S03	Student3	55
104	S01	Student4	70
105	S01	Student5	59
106	S02	Student6	71
107	S03	Student7	83
108	S01	Student8	91
109	S01	Student9	21

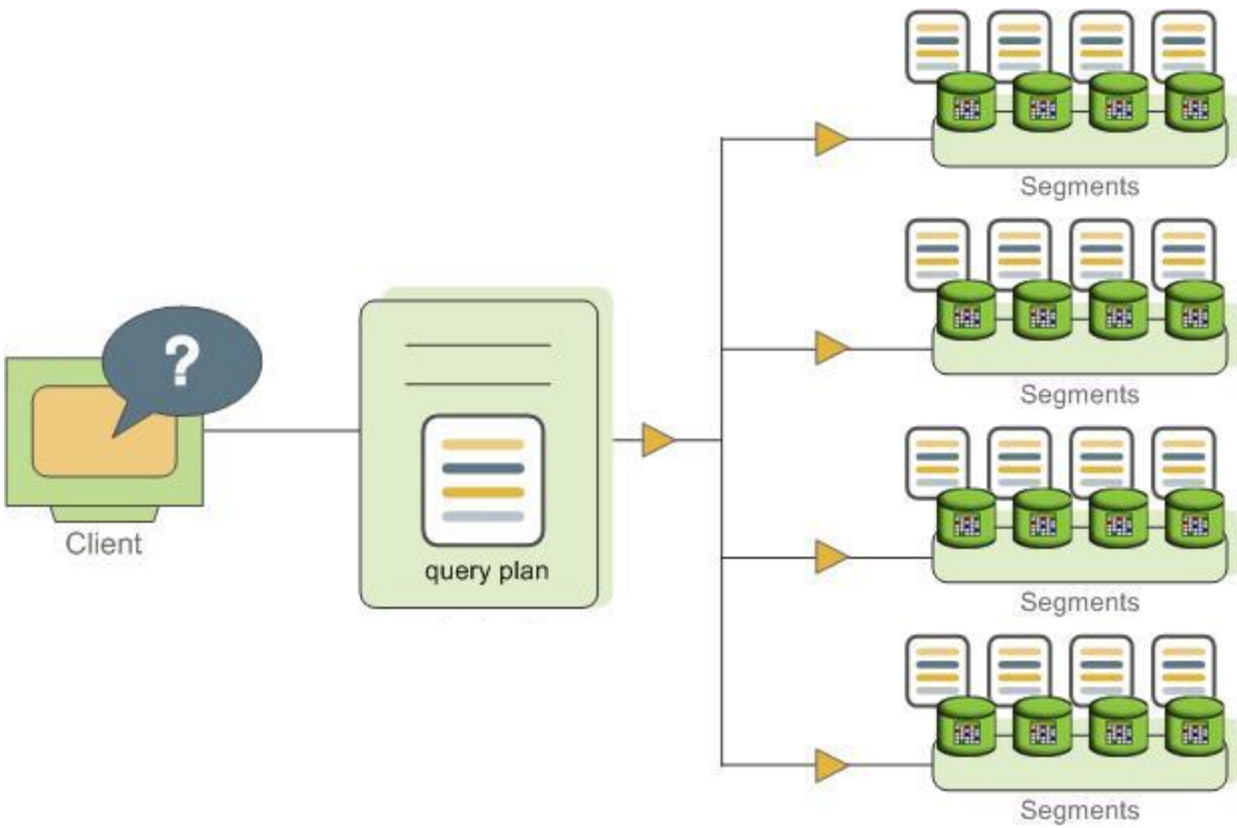


Data Access Times



Source: Greenplum Table Distribution and Best Practices, Greenplum table distribution and partitioning

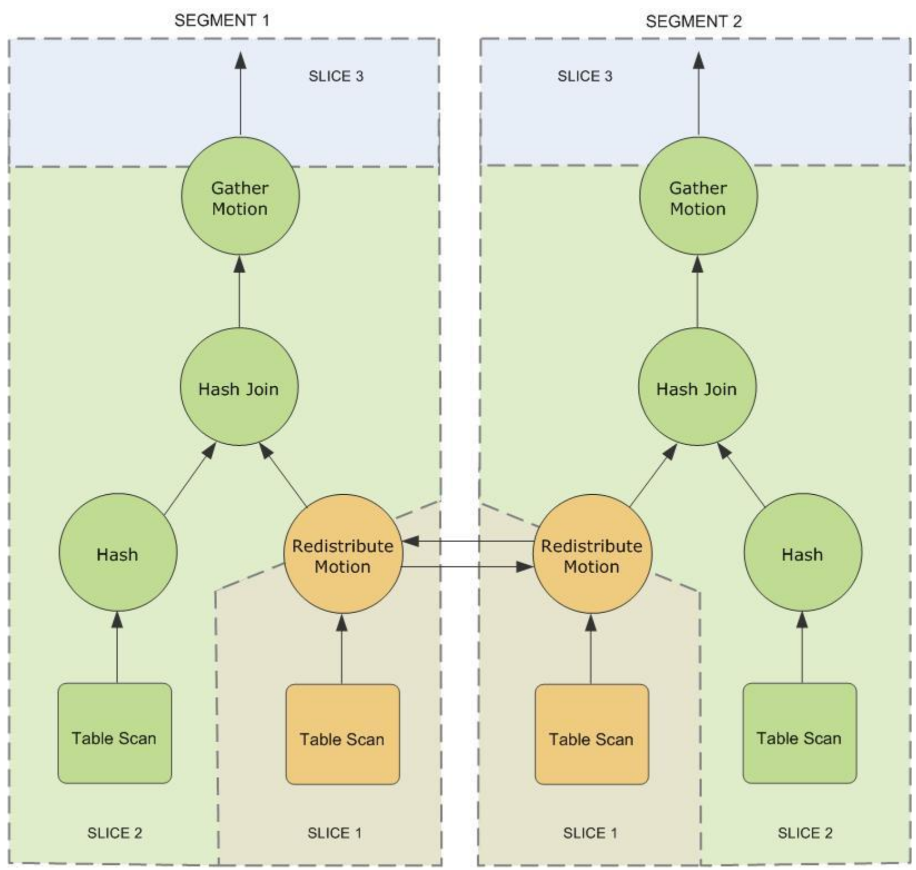
Coordinator Host Dispatches A Distributed Plan To Segments



Source: VMware Tanzu™ Greenplum® Documentation

Motions & Query Slice Plans

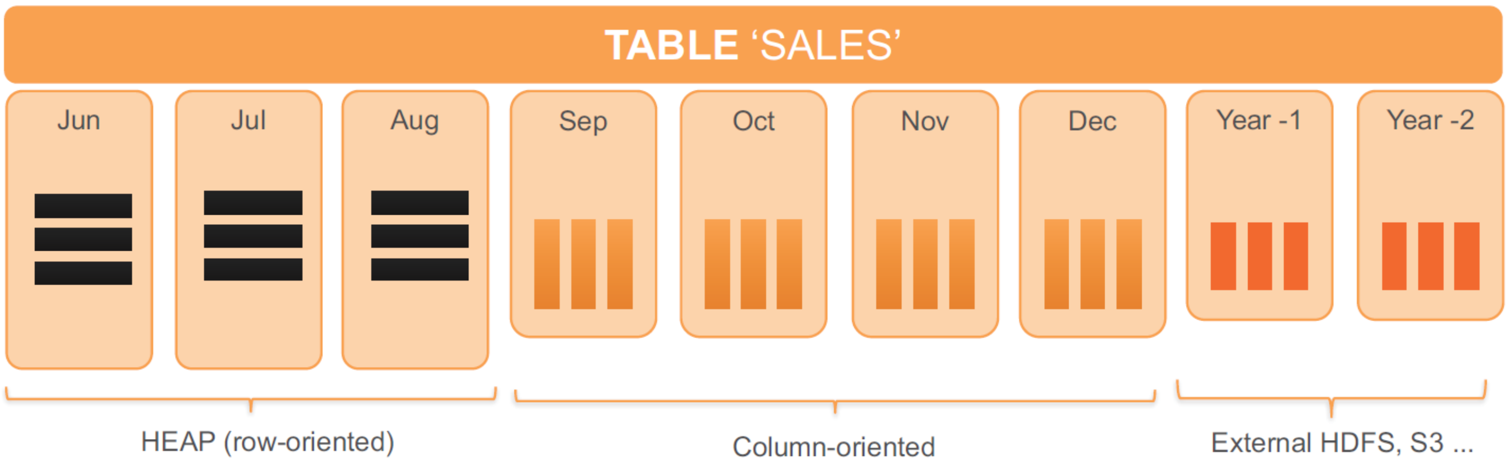
Motions involves moving tuples between segments



Source: VMware Tanzu™ Greenplum® Documentation

Table Storage Types

The query execution engine is agnostic to table storage type



Suitable for frequent updates and deletes. Use indexes for drill through queries.

Per column compression (zstd, gzip, quicklz, RLE with delta) and block size can be specified.

Less accessed, archived data. Supported formats include CSV, Binary, Avro, Parquet, etc.

Source: Lyu, Zhenghua et al. Greenplum: A Hybrid Database for Transactional and Analytical Workloads, SIGMOD '21: Proceedings of the 2021 International Conference on Management of Data.

Greenplum OLTP Optimizations

- Object Lock Optimizations
- Global Deadlock Detection (GDD)
- Distributed Transaction Management
- Resource Isolation

Object Lock Optimizations

Lock Mode	Level	Conflict Lock Level	Typical Statements
AccessShareLock	1	8	Pure SELECT
RowShareLock	2	7, 8	SELECT FOR UPDATE, UPDATE, DELETE
RowExclusiveLock	3	5, 6, 7, 8	INSERT
ShareUpdateExclusiveLock	4	4, 5, 6, 7, 8	VACUUM (not FULL)
ShareLock	5	3, 4, 6, 7, 8	CREATE INDEX
ShareRowExclusiveLock	6	3, 4, 5, 6, 7, 8	CREATE COLLATION
ExclusiveLock	7	2, 3, 4, 5, 6, 7, 8	REFRESH MATERIALIZED VIEW
AccessExclusiveLock	8	1, 2, 3, 4, 5, 6, 7, 8	ALTER TABLE

PURDUE UNIVERSITY®

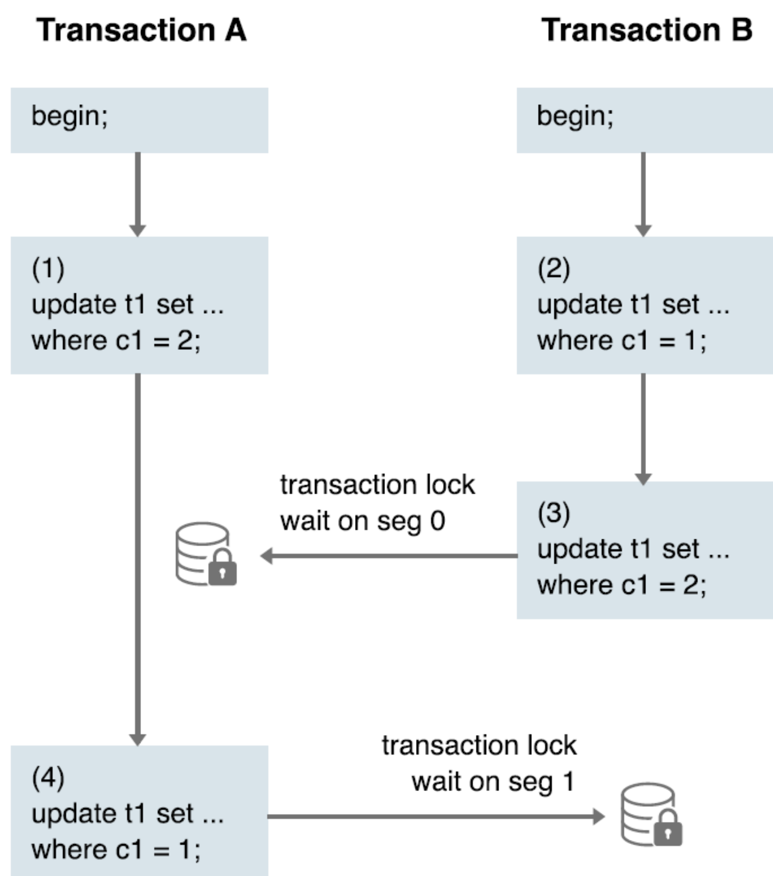
Source: Lyu, Zhenghua et al. Greenplum: A Hybrid Database for Transactional and Analytical Workloads, SIGMOD '21: Proceedings of the 2021 International Conference on Management of Data.

Global Deadlock Detection (GDD)

Transaction Operations in Time Order

Data Distribution

segment id	tuples (c1, c2)
0	(2, 2)
1	(1, 1)



Global Wait-for Graph

seg 0	B ⟶ A
seg 1	A ⟶ B

Source: Lyu, Zhenghua et al. Greenplum: A Hybrid Database for Transactional and Analytical Workloads, SIGMOD '21: Proceedings of the 2021 International Conference on Management of Data.

Distributed Transaction Management

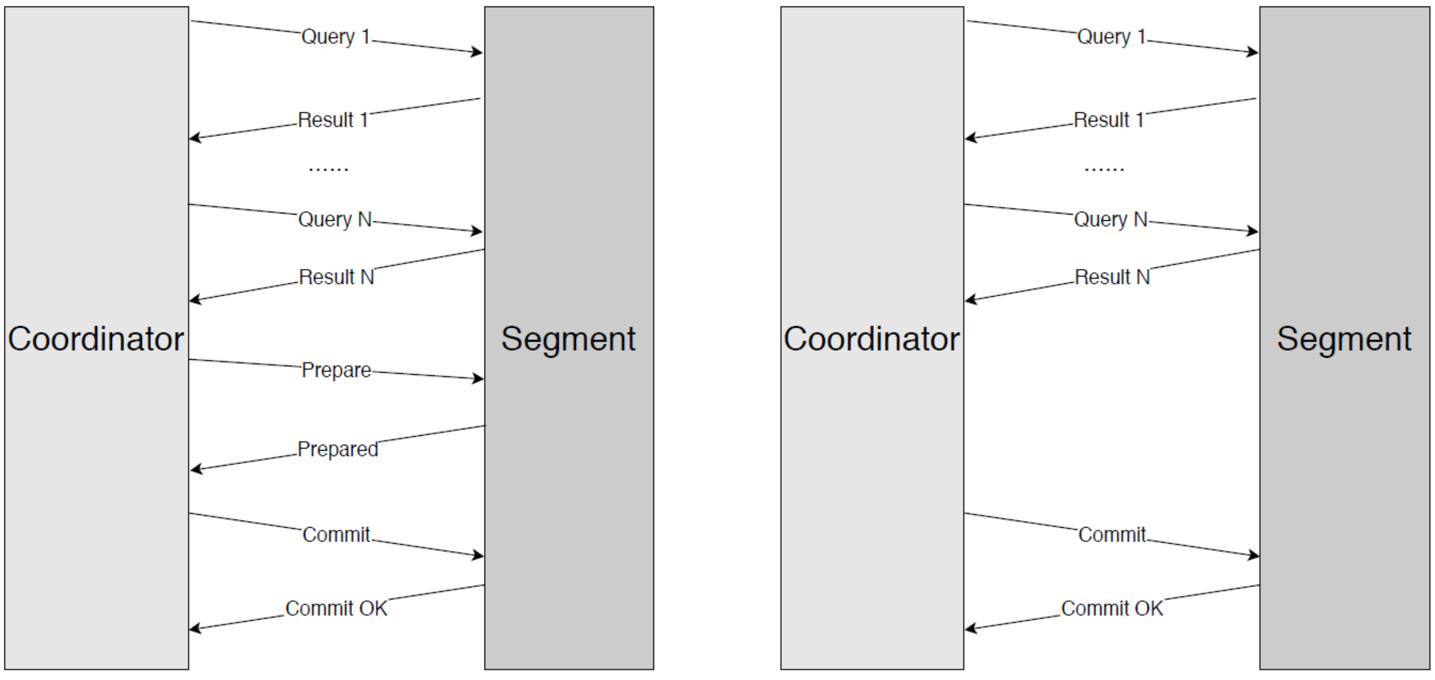
- Each transaction is assigned a unique transaction ID (XID), an incrementing 32-bit value.
- When a transaction inserts a row, the XID is saved with the row in the *xmin* system column.
- When a transaction deletes a row, the XID is saved in the *xmax* system column.
- Updating a row is treated as a delete and an insert, so the XID is saved to the *xmax* of the current row and the *xmin* of the newly inserted row.
- The *xmin* and *xmax* columns, together with the transaction completion status, specify a range of transactions for which the version of the row is visible.

Distributed Transaction Management (Continued)

- Multi-statement transactions also record which command within a transaction inserted a row (*cmin*) or deleted a row (*cmax*).
- The segments maintain a mapping of distributed transaction IDs with their local XIDs.

One-Phase Commit Protocol

- One-phase commit is an optimization for transactions that update data on exactly one segment.



(a) Two phase commit (b) One phase commit

Source: Lyu, Zhenghua et al. Greenplum: A Hybrid Database for Transactional and Analytical Workloads, SIGMOD '21: Proceedings of the 2021 International Conference on Management of Data.

Resource Isolation

- CPU isolation is implemented based on Linux control groups (*cgroup*).
- Memory isolation is implemented based on the memory management module *Vmemtracker.*
- For CPU resources, we assign more CPU rate limit to the transactional resource group, since transactional queries are short and sensitive to query latency.
- Higher memory limit is assigned to the analytical resource group to allow analytical queries to use more memory and to avoid spilling to disk excessively.
- Future work is going on to introduce a workload prediction module, which allows a query to use more memory when the prediction of incoming workload is heavy, even when not set to do so.

Performance Comparison

OLTP Performance (TPC-B Like Benchmark)

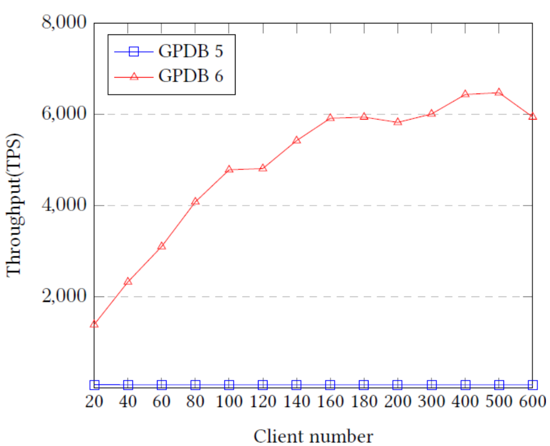


Figure 12: TPC-B Like Benchmark Result

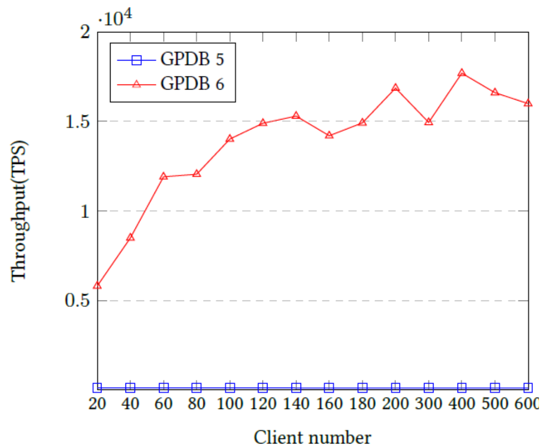


Figure 14: Update Only Workload Result

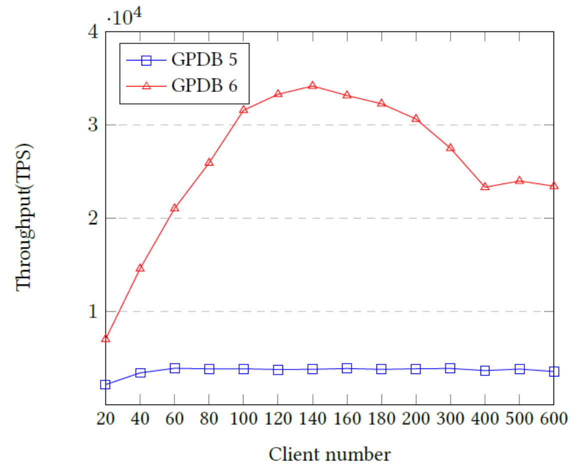


Figure 15: Insert Only Workload Result

Source: Lyu, Zhenghua et al. Greenplum: A Hybrid Database for Transactional and Analytical Workloads, SIGMOD '21: Proceedings of the 2021 International Conference on Management of Data.

Performance Comparison

HTAP Performance (CH-benCHmark)

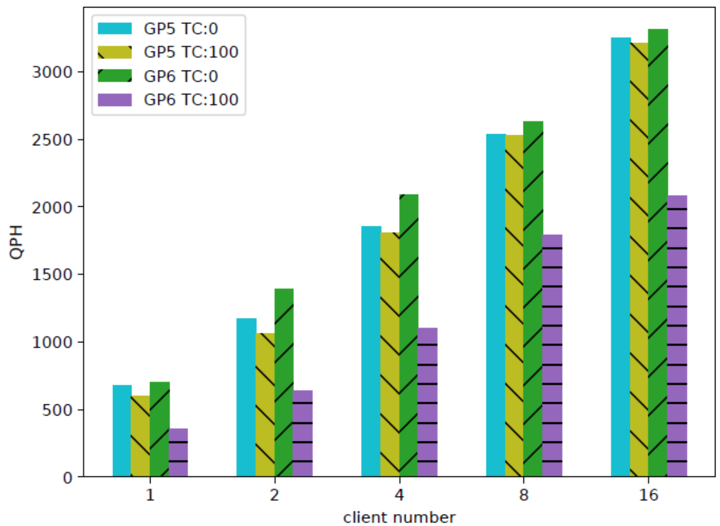


Figure 16: OLAP performance for HTAP workloads

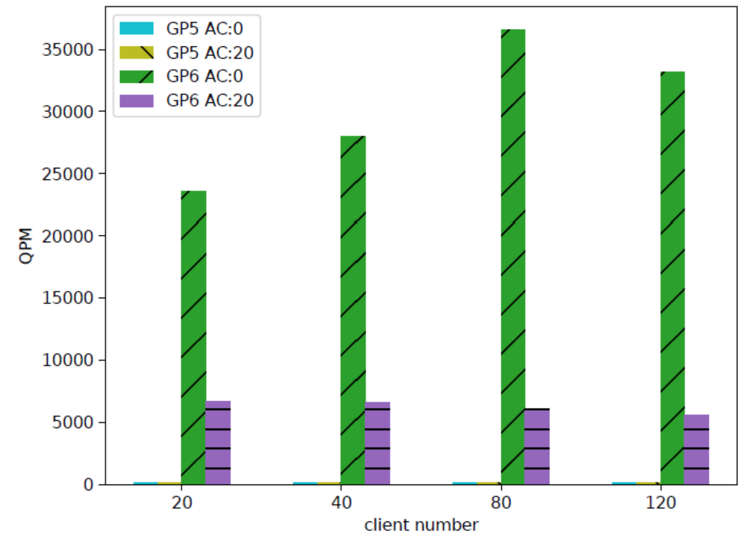


Figure 17: OLTP performance for HTAP workloads

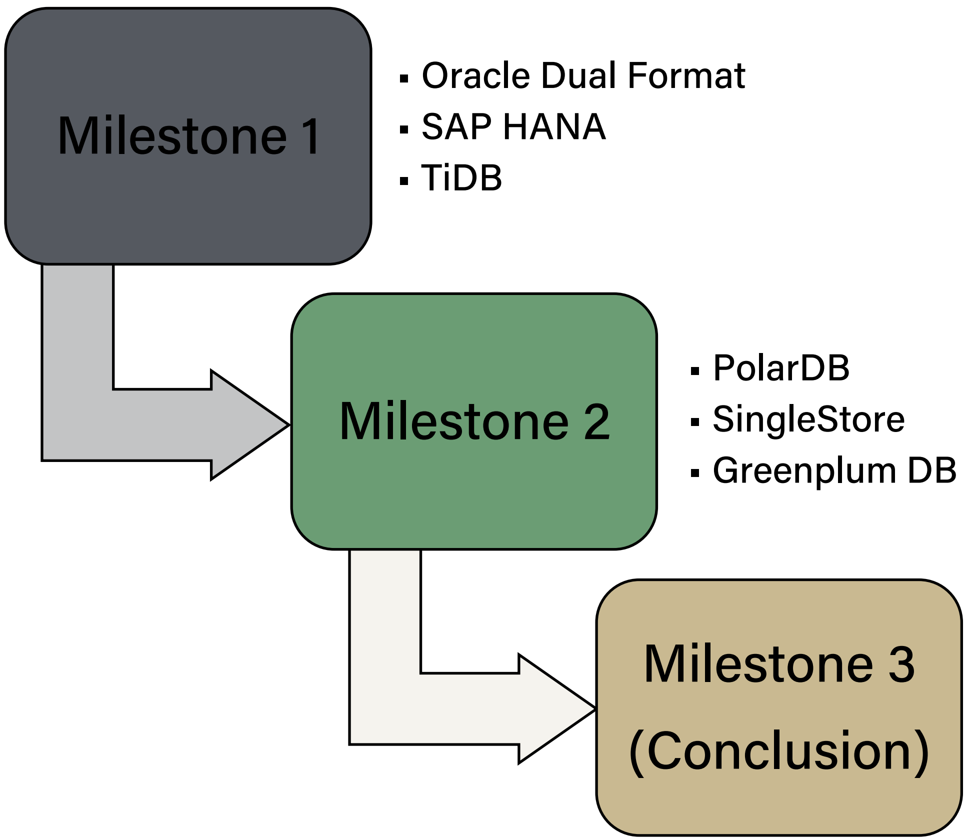
Source: Lyu, Zhenghua et al. Greenplum: A Hybrid Database for Transactional and Analytical Workloads, SIGMOD '21: Proceedings of the 2021 International Conference on Management of Data.

Project Plan

- Study HTAP systems with diverse architectures (e.g. data organisation and synchronization)
  - Compare and Contrast such systems based on their design decisions
  - Assess the strengths and weakness of these systems (in terms of AP/TP throughput, scalability, among others)
- Study the systems that have evolved from OLTP to HTAP (e.g. MemSQL, IBM dashDB, and more)
- Summarize all the systems studied and create a taxonomy based on design and performance

Project Timeline

Survey Project Timeline



- Oracle Dual Format
- SAP HANA
- TiDB

- PolarDB
- SingleStore
- Greenplum DB

- Study of the next 3 HTAP systems
- Summarize all the systems studied and create a taxonomy based on design & performance

THANK YOU