TiQuE: Improving the Transactional Performance of Analytical Systems for True Hybrid Workloads

Nuno Faria[1], José Pereira[1], Ana Nunes Alonso[1], Ricardo Vilaça[1], Yunus Koning[2], and Niels Nes[2,3]
[1] INESCTEC & University of Minho | [2] MonetDB Solutions | [3] CWI

VLDB 2023

Motivation

- Hybrid Transactional-Analytical Processing (HTAP) systems are designed to efficiently handle transactional (OLTP) and analytical (OLAP) workloads.
- Each current approach, however, has different limitations:
  - **Two specialized systems + replication** – delay between OLAP and OLTP data; higher operational and maintenance costs of managing multiple systems;
  - **Same system, multiple data copies** – possible delay between OLAP and OLTP data; higher storage requirements;
  - **Same system, same data** – complex implementation and maintenance.

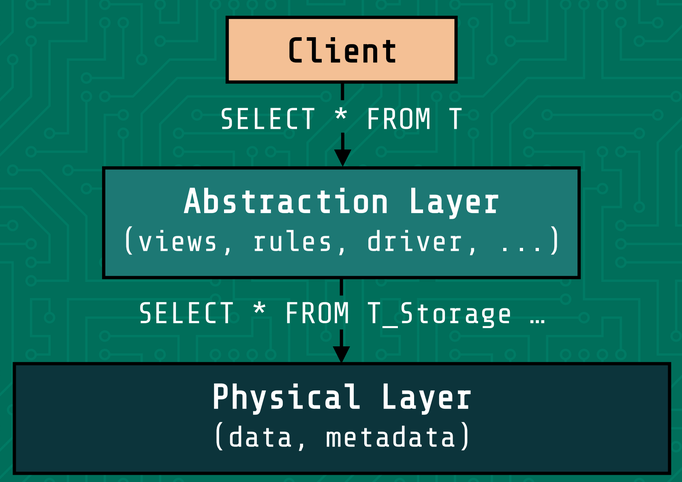

Figure 1: TiQuE's architecture.

Transactions in the Query Engine

- TiQuE targets HTAP with the same data. However, **transactional isolation is implemented in the query engine** through high-level queries, instead of hardcoding it in the storage engine, easing the implementation effort.
- Its adaptable to **custom isolation criteria**, depending on the application needs.
- It allows transactional operations to be **optimized with the workload**, thus being tunable and able to utilize techniques such as vectorization and parallelism.

Schema

- **Storage tables** – store stable data; same schema as the original tables.
- **Cache tables** – store uncommitted/recently committed data (with metadata).
- **Txn table** – stores information about transactions, such as timestamps.
- **TXID/STS/CTS sequences** – assign identifiers and timestamps.

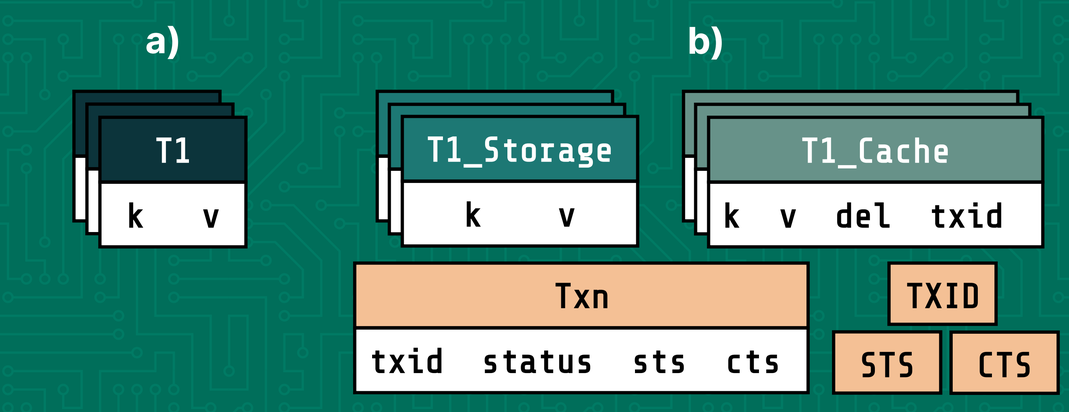

Figure 2: Original application schema (a) converted to TiQuE (b).

Operations

- TiQuE translates **begin**, **read**, **write**, and **commit** operations to use the target isolation.
- This translation can be done with a client-side driver or with views and rules.

```
SELECT k, v
FROM (
  SELECT *, rank() OVER (PARTITION BY k
      ORDER BY cts DESC NULLS FIRST) AS rk
  FROM (
    (SELECT k, v, false AS del, 0 AS cts
     FROM T1_Storage)
    UNION ALL
    (SELECT k, v, del, cts
     FROM T1_Cache C
     JOIN Txn ON Txn.txid = C.txid
     WHERE (Txn.status = 'T' OR Txn.txid = MY_TXID)
       AND (cts <= MY_STS OR cts IS NULL))
  ) T1
) T2
WHERE rk = 1 AND NOT del;
```

Figure 3: Possible snapshot computation in TiQuE.

Checkpoint and Recovery

- **Checkpoint** - periodically, stable data are moved from the cache to the respective storage table, to keep overhead low.
- **Recovery** – when the system restarts after a crash, the transactional metadata are updated to ensure consistency.

Selected Results

- Based on a prototype implemented on top of MonetDB, a state-of-the-art SQL analytical system.
- The OLTP tests were evaluated with TPC-C, while the OLAP were evaluated with CH-benCHmark.

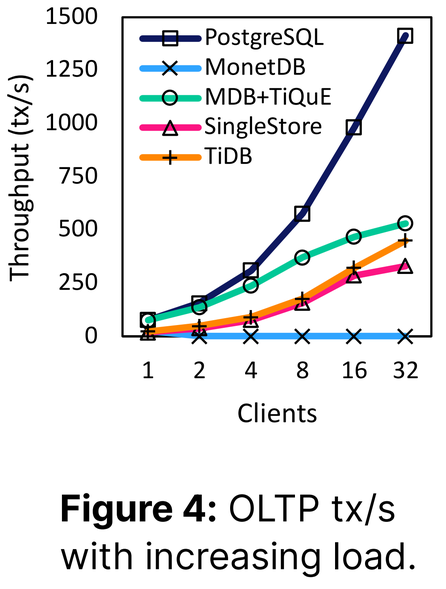

Figure 4: OLTP tx/s with increasing load.

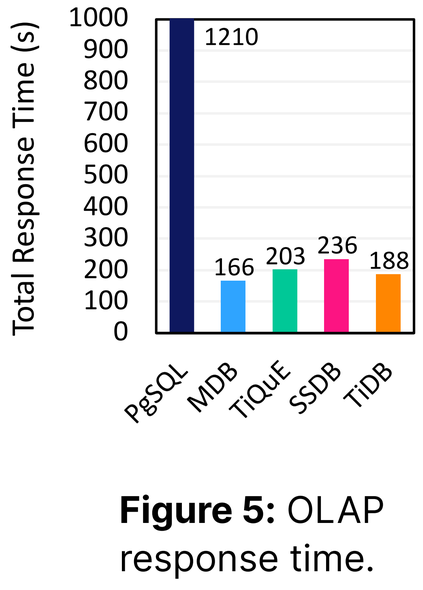

Figure 5: OLAP response time.

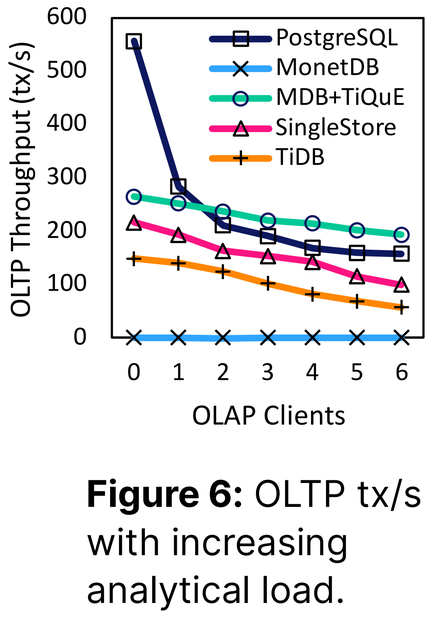

Figure 6: OLTP tx/s with increasing analytical load.

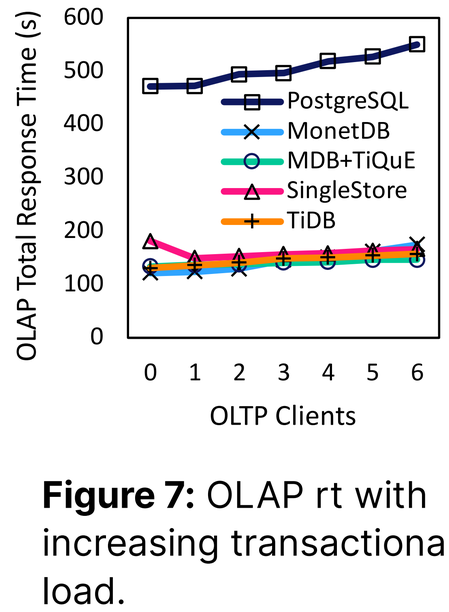

Figure 7: OLAP rt with increasing transactional load.

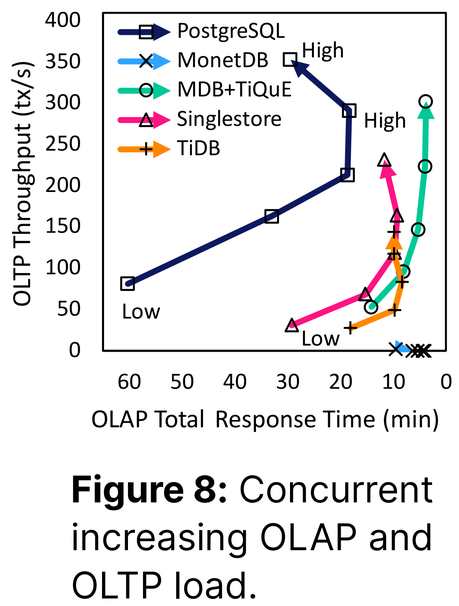

Figure 8: Concurrent increasing OLAP and OLTP load.

FCT Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

This work is financed by National Funds through the Portuguese funding agency, FCT – Fundação para a Ciência e a Tecnologia, within project LA/P/0063/2020.

Scan to read the full paper.