

Two Birds With One Stone: Designing a Hybrid Cloud Storage Engine for HTAP

Tobias Schmidt, Dominik Durner, Viktor Leis, Thomas Neumann

Technische Universität München, CedarDB

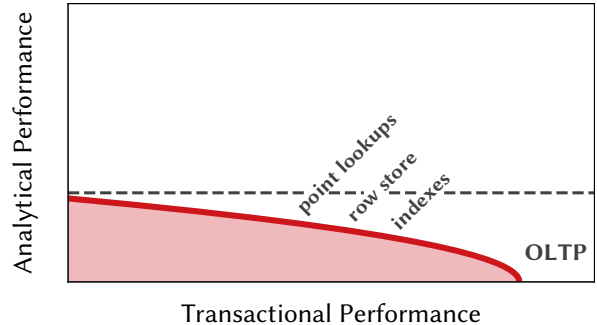
26.08.2024



Increasing demand for real-time analytics on up-to-date data.

Increasing demand for real-time analytics on up-to-date data.

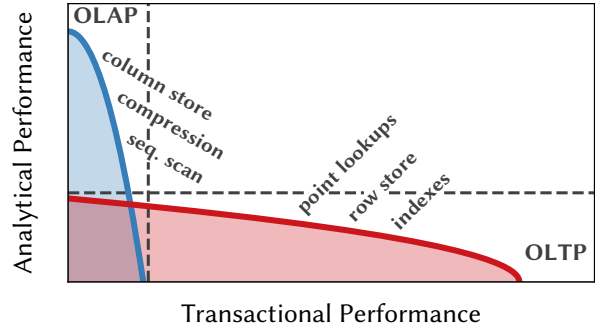
OLTP systems fail to analyze large data sets efficiently.



Increasing demand for real-time analytics on up-to-date data.

OLTP systems fail to analyze large data sets efficiently.

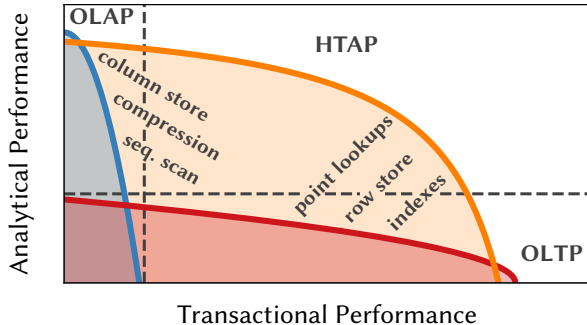
OLAP systems fail to provide high transaction rates.



Increasing demand for real-time analytics on up-to-date data.

OLTP systems fail to analyze large data sets efficiently.

OLAP systems fail to provide high transaction rates.



⇒ Colibri combines OLAP and OLTP architectures to process analytical and transactional workloads.

Fast analytics through *columnar storage* and *vectorized processing*.

High transaction rates through *index structures* and *inexpensive point accesses*.

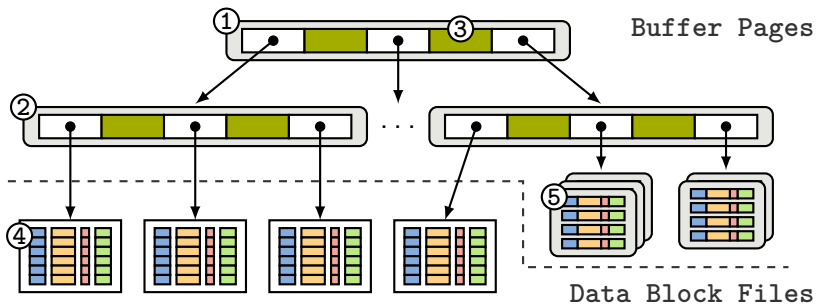
Fast analytics through *columnar storage* and *vectorized processing*.

High transaction rates through *index structures* and *inexpensive point accesses*.

Core Techniques:

- ▶ **Hybrid column-row store**
- ▶ **Bandwidth-optimized table scans**
- ▶ **Lightweight in-memory MVCC**
- ▶ Bypassing the write-ahead log
- ▶ Custom compressed data format
- ▶ Swip-based buffer manager
- ▶ Optimistic lock coupling

Hybrid column-row store:



① B⁺-tree (inner node)

② B⁺-tree (leaf node)

③ Row Id (B⁺-tree key)

④ Compress Data Blocks

⑤ Uncompressed Data Pages

INSERT INTO ...

- ▶ small change set: new tuples added to the hot section in row format
- ▶ large change set: compress tuples upfront and add them to the cold section

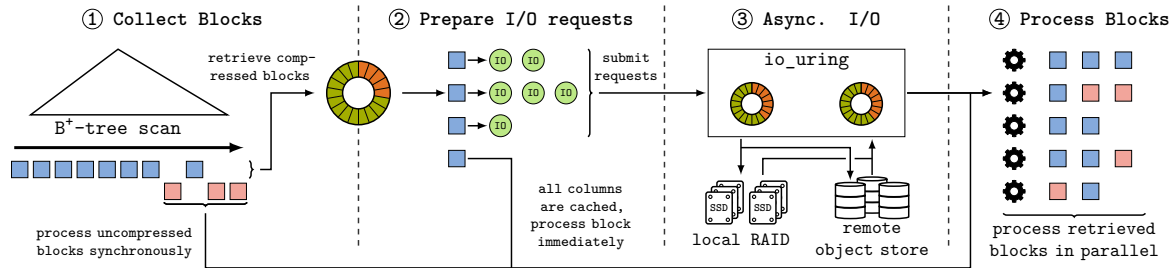
DELETE FROM ...

- ▶ uncompressed data page: mark tuple as invisible on the page
- ▶ compressed data blocks: additional bitset page to mark tuples as deleted

UPDATE ... SET ...

- ▶ uncompressed data page: in-place update
- ▶ compressed data blocks: delete tuple and insert with updated columns

Asynchronous table scan:



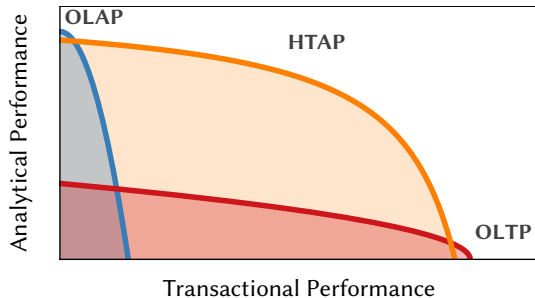
- ▶ Colibri fetches compressed data blocks asynchronously using `io_uring`
- ▶ up to 50 GB/s on SSDs, 100 Gbit/s on Amazon S3

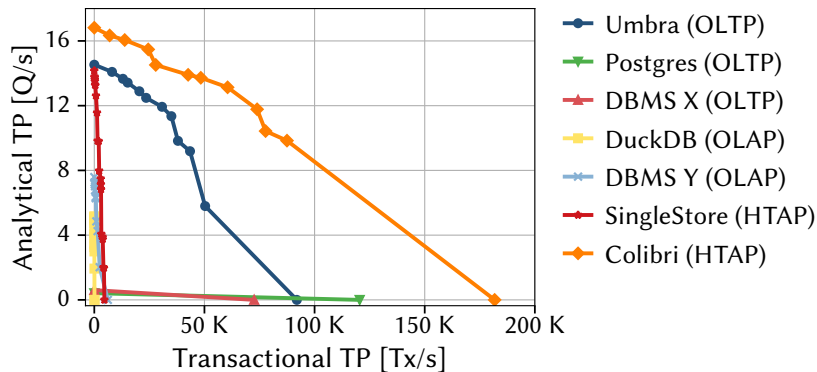
Throughput Frontier:

- ▶ Workloads consist of transactions and analytical queries.
- ▶ Run with different combinations of analytical and transactional clients.
- ▶ Skyline of measured throughputs.

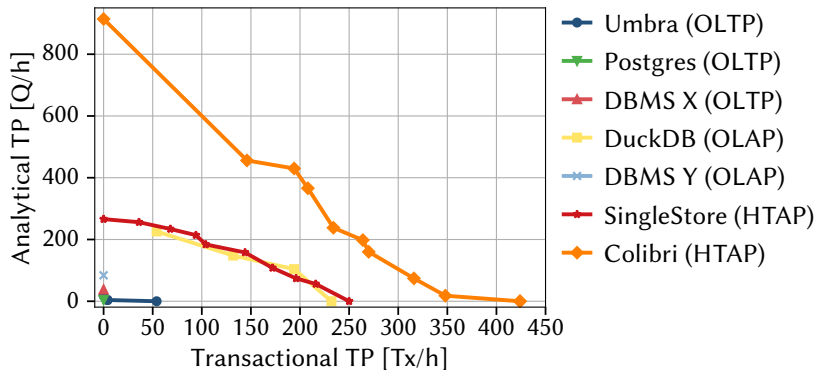
Setup:

- ▶ 2x AMD EPYC 7713
- ▶ 1024 GB of RAM
- ▶ 8x PM9A3 SSDs in RAID0





Colibri hybrid column-row store outperforms specialized OLAP and OLTP architectures.



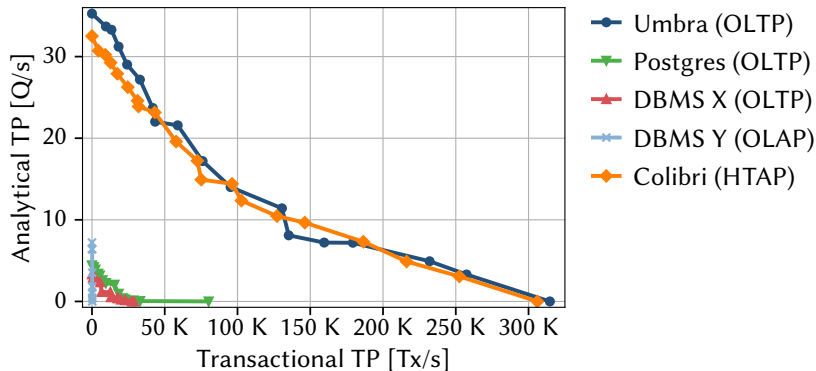
Columnar access and compression are key for fast analytics on large datasets.

Evaluation

OLTP: CH-benCHmark



UMBRA



Transactional performance is on par with state-of-the-art OLTP systems.

Colibri versions	1 client	32 clients
column store	3 Tx/s	45 Tx/s
+ secondary indexes	465 Tx/s	2 685 Tx/s
+ row store	5 404 Tx/s	94 644 Tx/s
+ in-place updates	21 508 Tx/s	288 719 Tx/s

Secondary indexes and uncompressed data pages are decisive for high transaction rates.

Colibri's hybrid column-row store bridges the gap between analytical and transactional systems.

Colibri's hybrid column-row store bridges the gap between analytical and transactional systems.

- ▶ Compressed data blocks allow for vectorized processing, asynchronous data retrieval, and columnar access.
- ▶ Uncompressed data pages enable fast point lookups and in-place updates.
- ▶ Colibri combines the two formats in a B^+ -tree.