Microsoft

Microsoft Research Faculty Summit 2013

Microsoft

HyPer - A combined OLTP and OLAP engine

Thomas Neumann
Technische Universität München

Motivation

Traditionally, DBMSs either good at OLTP or good at OLAP

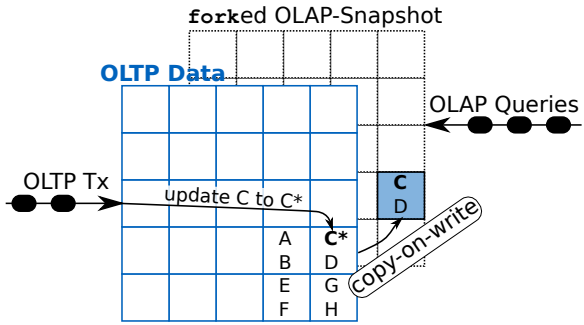
- OLTP
  - high rate of small/tiny transactions
  - high locality in data access
  - update performance is critical
- OLAP
  - few, but long running transactions
  - aggregates large parts of the database
  - must see a consistent database state the whole time

Leads to conflicts. Traditional solutions like 2PL would block OLTP.
But: main-memory database have new options.

Transaction Support

HyPer isolates long-running transactions
(e.g., OLAP) using virtual memory snapshots.

- "copy" the database on demand
- the MMU/OS keeps track of changes
- only the working set is replicated
- snapshots remains constant
- very little overhead
- optimistic CC for back merge



Extremely fast execution model, no overhead for locking etc.
Supports OLTP and OLAP simultaneously.

Execution Model

Main memory so fast that CPU usage becomes a problem

- classical iterator model fine for disks, but not so for main memory
- many branches, bad locality (code and data)
- fine when waiting for disk, hurts in main memory

Principles of HyPer's execution strategy

- touch data as rarely as possible (avoid memory "I/O")
- prefer tight worker loops instead of spread out control logic

Less of an issue for OLTP, but crucial for OLAP.
And even OLTP feels CPU usage.

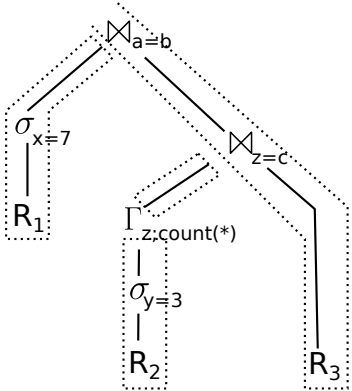
Data Centric Execution

Ideally, process pipeline fragments in tight loops

1. load data from the source pipeline breaker into CPU registers
2. perform all pipelining operations
3. materialize into the next pipeline breaker

Minimized memory accesses and produces compact code

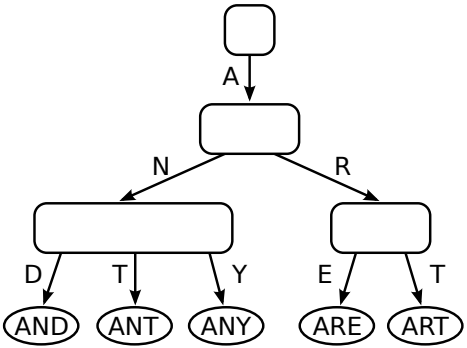
- runtime native code generation using LLVM
- no interpretation overhead
- matches performance of hand-written code



Indexing

OLTP is dominated by index accesses.

- hash table
  - \+ very fast, (nearly) direct access
  - – no range queries
  - – non-unique indexes difficult
  - – hash table growth
- trees
  - \+ range queries
  - – tree depths
  - – compare+branch is slow
- radix tree
  - \+ direct jumps, no comparisons, still range queries
  - – space utilization
  - – (potentially) tree depth



HyPer uses a heavily tuned radix tree as default index. Compact, fully featured, and fast.

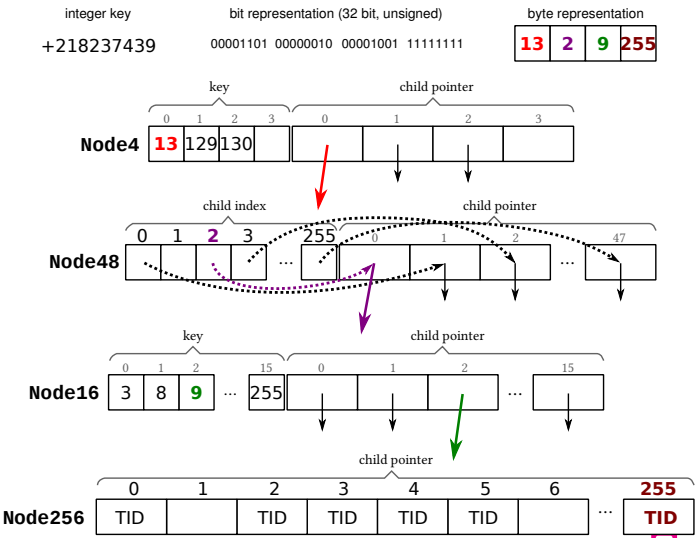
Adaptive Radix Tree

adaptive sizes

- adapts to data distribution
- avoids underfull nodes
- fixed bound on space per entry (regardless of key size)!

techniques omitted to keep the example readable

- prefix compression
- path compression
- lazy expansion



Excellent behavior for a wide range of uses cases.

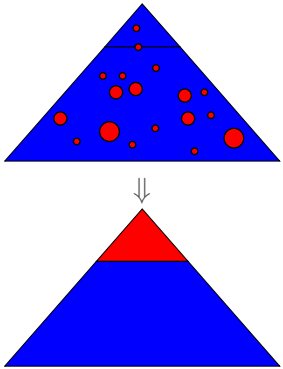
Compaction

Databases can be huge, but OLTP working set usually modest.

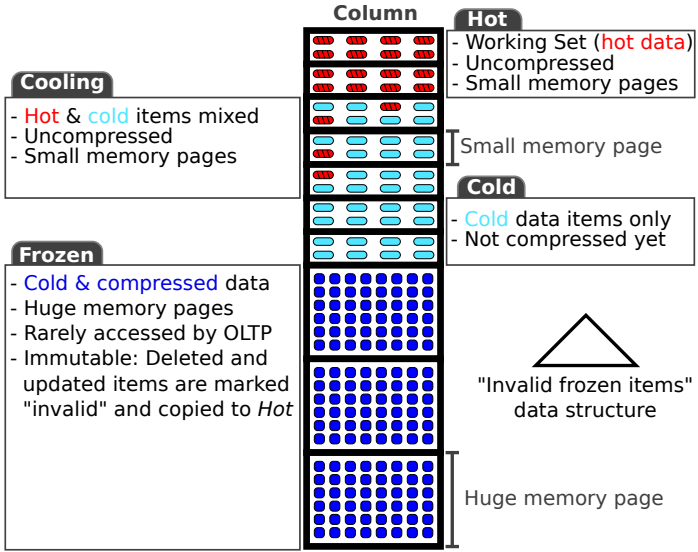
- old data is rarely changed
- changes concentrate in small parts of the database
- not necessarily physically near, though

Compaction reduces the spread of the working set

- good for locality (and copy-on-write)
- more aggressive storage for read-mostly data
- huge pages, compression, etc.
- or even disk



Hot/Cold-Partitioning for Compaction



What to expect from a combined OLTP and OLAP engine

Some numbers to get an impression.
64GB server, full ACID with serializability, one thread for OLTP and OLAP each.

TPC-C 12 warehouses, no wait time
138,000 transactions per second

TPC-H SF=10, executing all 22 queries
14,2 seconds

TPC-C+H simultaneously H queries adapted for C, OLAP on OLTP data
122,000 transactions per second, minimal impact on OLAP

Excellent performance. On OLTP and OLAP, and both simultaneously!

Conclusion

Main-memory changes a lot for database systems

- more than a fast disk
- allows for techniques that are not possible with disks
- indexing and execution different than in disk-based systems

HyPer demonstrates that unifying OLTP and OLAP is possible now

- excellent performance both in OLTP and OLAP
- concurrent execution of both OLTP and OLAP has only modest effect on OLTP
- full ACID, SQL-92, no partitioning restrictions

`http://www.hyper-db.com`

(live demo should be online soon)