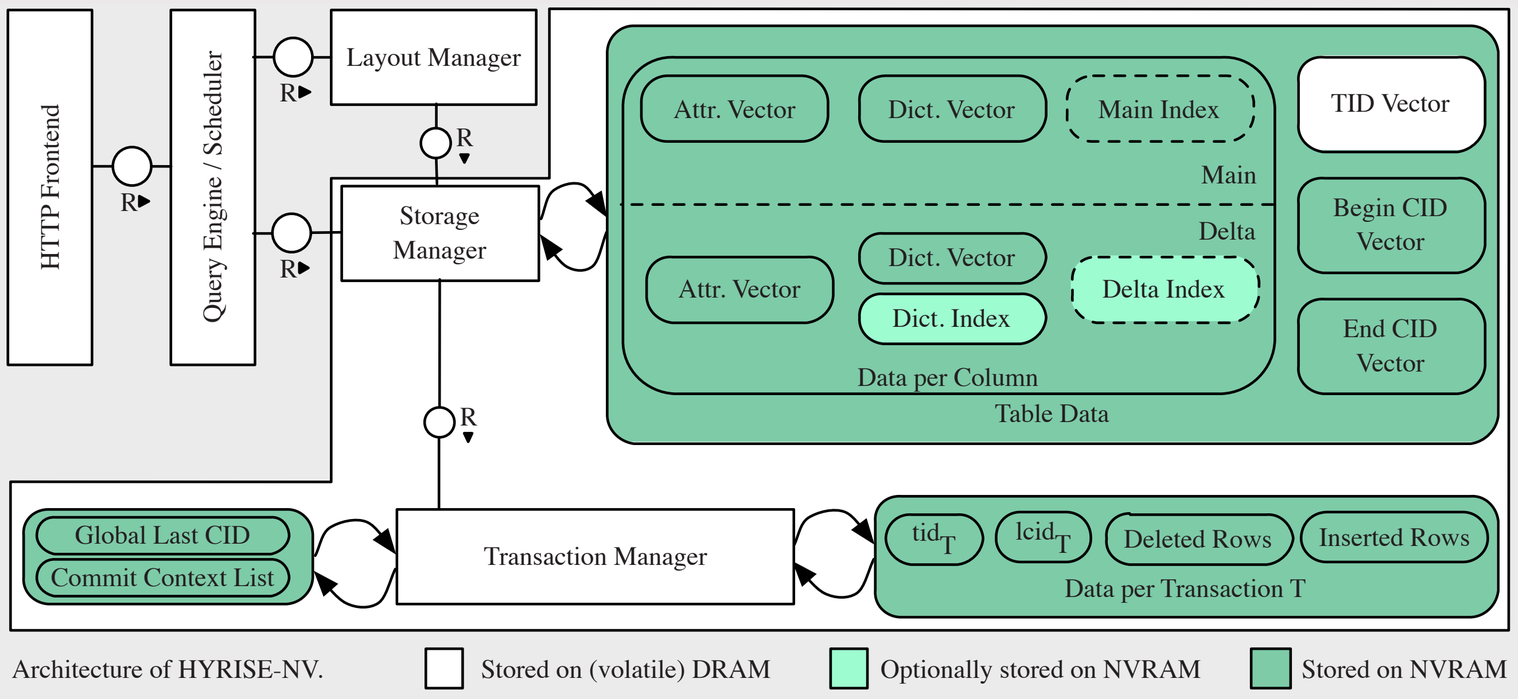
Using non-volatile RAM for inherent persistence and fast recovery of an in-memory database

- Writing database checkpoints and logs to disk makes DBMS architectures more complex
- Recovering takes time in which the DBMS is not fully functional
- New data structures on non-volatile RAM (NVRAM) are persistent and require only little recovery
- Hyrise-NV recovers in ~100ms, virtually independent of the database's size, excluding a reboot of the operating system



Architecture of HYRISE-NV.

ARCHITECTURE

What architecture is this based on? Tables are in a dictionary-encoded, columnar layout using a main-delta architecture and MVCC. Main and delta have attribute vectors with value ids pointing to the dictionary. Main dictionaries are sorted for fast reads. Inserts use a write-optimized delta. Delta dictionaries are unsorted.

PERSISTENCE

How to guarantee ACID conformance? In addition to the attribute vector and the dictionaries, the MVCC information is stored on NVRAM. Thus, deleted and uncommitted rows can be identified. To speed up the rollback, additional data is stored per transaction. For instant recovery, indexes are persisted on NVRAM but can also be rebuilt on restart.

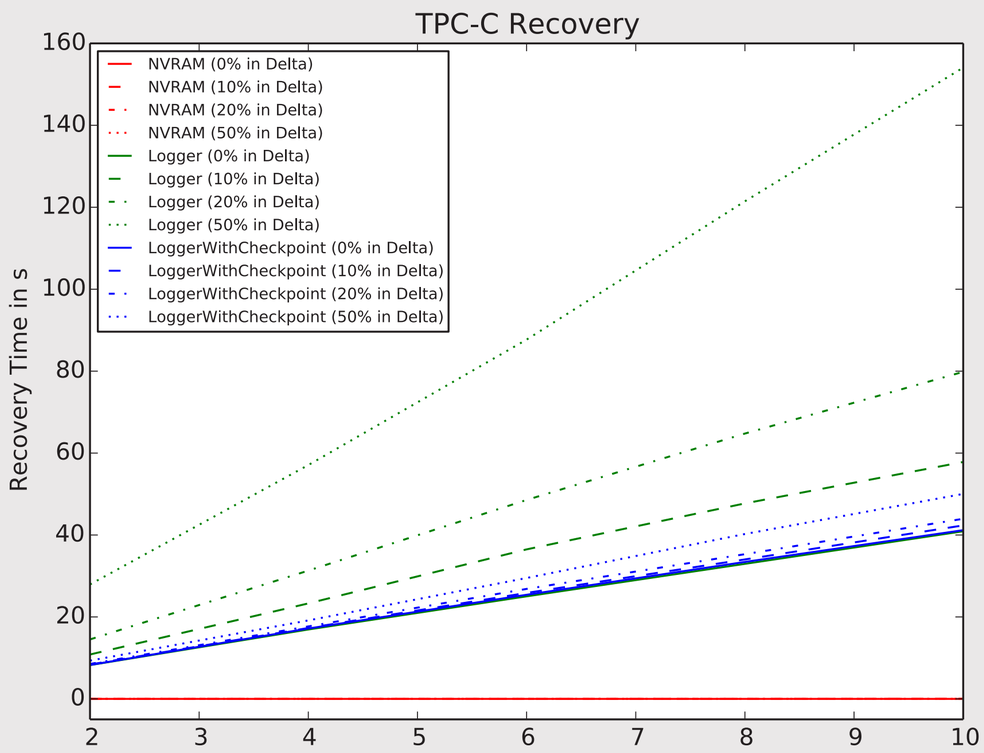
DATA STRUCTURES

What data structures can be used for persistence? How are they adapted?

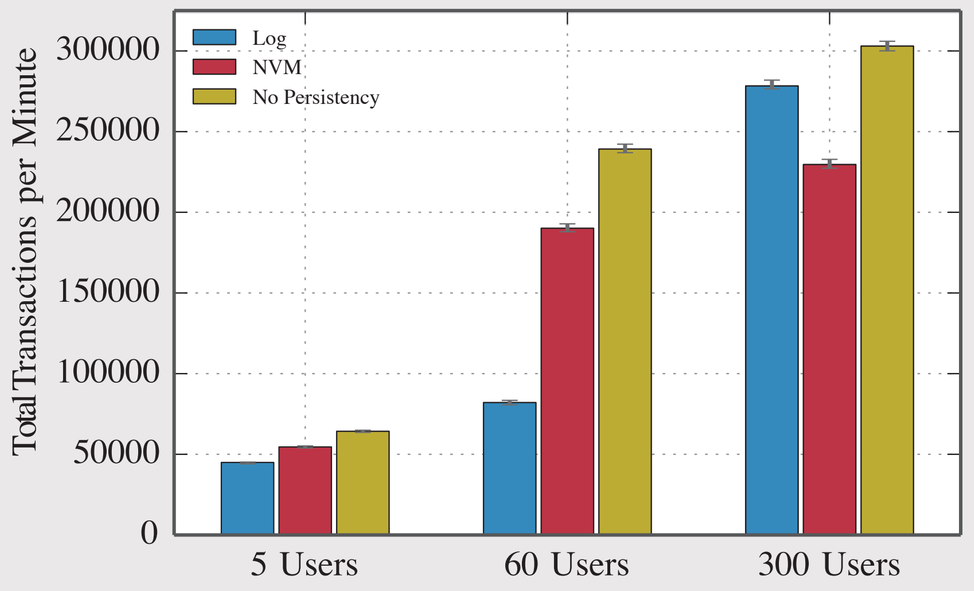
- Attribute, dictionary, and MVCC vectors: stored in contiguous NVRAM
- Indexes: new, optimized version of the STX B+-Tree with multiversioning. Alternatively, concurrent hashmap adapted for NVRAM (the NVC-Hashmap)
- Fences and cache line flushes guarantee that data reaches NVRAM

RECOVERY

How to recover in case of a crash? No logging and only minimal cleanup is needed: For recovery, uncommitted transactions are reverted in the MVCC vectors based on the Deleted / Inserted Rows information. Unfinished operations on the trees are rolled back.



Recovery speed of different persistency mechanisms for n GB of data (less is better); NVRAM simulated



Runtime performance depending on number of users (more is better); Hyrise-NV is bound by performance of flushes NV

Project Members

Hasso Plattner Institute: David Schwalb, Martin Faust, Markus Dreseler, Tim Berning
NetApp: Girish Kumar BK, Anusha S, Adolf Hohl, Gaurav Makkar, Parag Deshmukh

NetApp™

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam