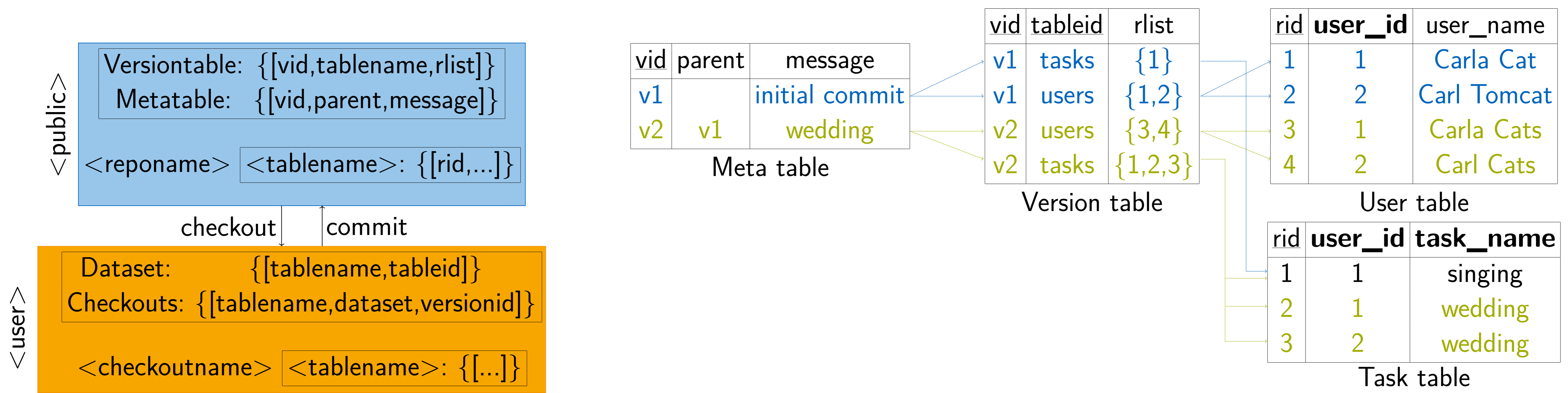


Versioning in Main-Memory Database Systems

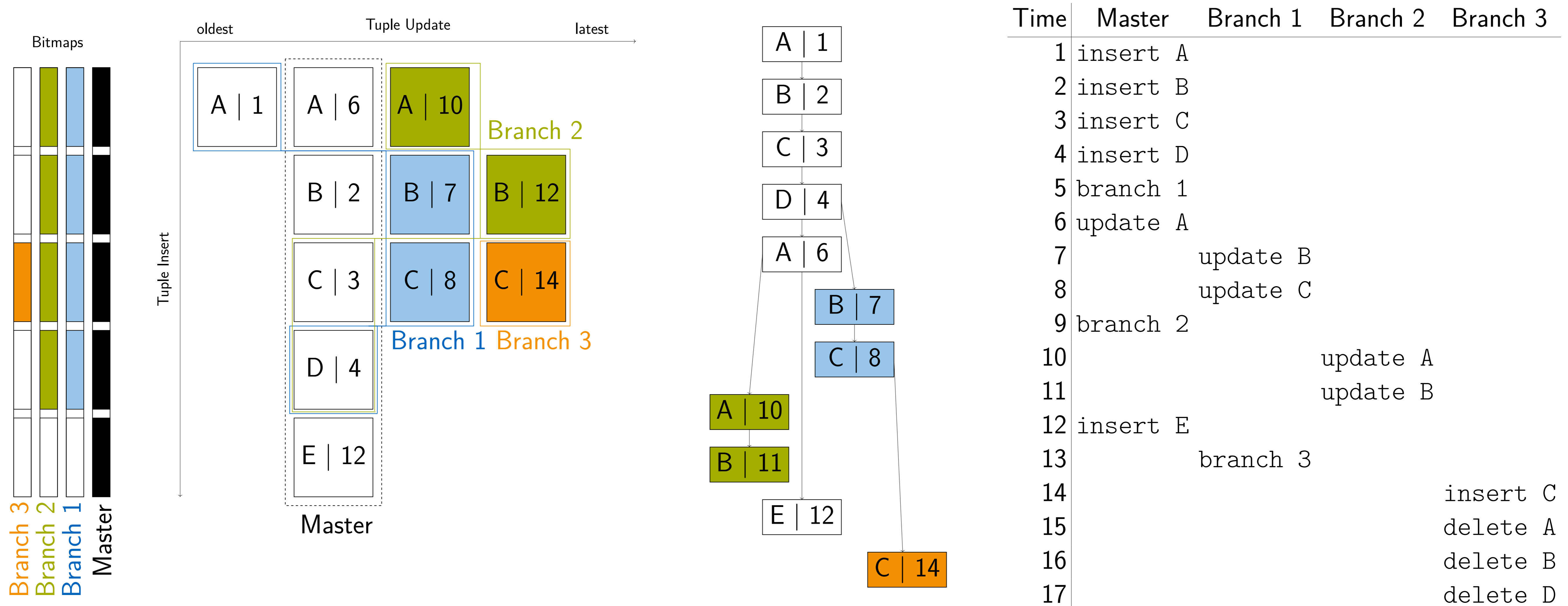
Maximilian E. Schüle, Lukas Karnowski, Josef Schmeißer, Benedikt Kleiner, Alfons Kemper, Thomas Neumann
 {m.schuele, lukas.karnowski, josef.schmeisser, benedikt.kleiner}@tum.de, {kemper, neumann}@in.tum.de

MusaeusDB: Versioning On Top of Relational Database Systems



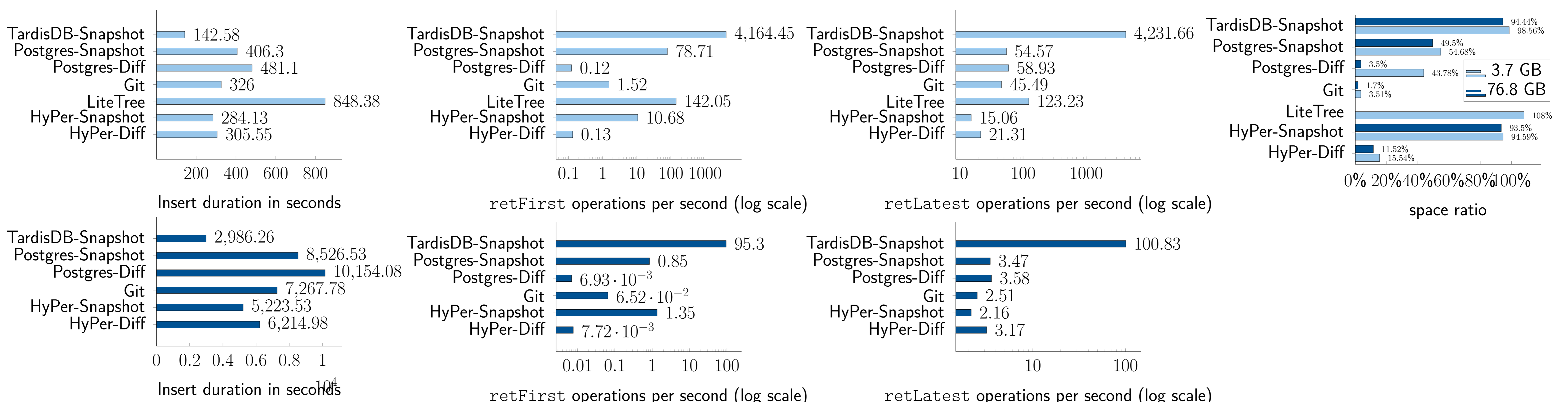
Distinction between global and local (user) space: The global space maintains a separate name-space for each repository, relations can be checked out for modifications in the user's namespace. Schema: Version table and meta table for managing the commits on the left; tables containing the data on the right; the record identifier (rid) serves as a key for every tuple.

TardisDB: Version Control Inside a Main-Memory Database System



Adaption of multi-version concurrency control for versioning (left): bitmaps for each branch indicate the included tuples; an insert increases the size of all bitmaps. Updates in the master branch are handled in place with a pointer to the previous version, updates from other branches are prepended. Tuples receive a unique timestamp, their colour indicates the creator branch. Descendance tree (middle) determines the tuple visibility for the corresponding history (right).

TardisBenchmark and Evaluation



The benchmark allows insertion or retrieval queries (retrieval of either the first or latest inserted page) and stores for each page either a whole snapshot or just the changes (differences). The first three rows show the performance of insertion queries and the retrieval of the first and latest page version. The upper part used a small dataset (3.7 GB), the lower one a bigger dataset (76.8 GB). Space consumption (last row).



gitlab.db.in.tum.de/tardisDB



European Research Council
 Established by the European Commission