



## Development of B-epsilon Tree Object Store

The [Be-Tree Storage Stack](https://github.com/Niliix007/betree_storage_stack) ([https://github.com/Niliix007/betree\\_storage\\_stack](https://github.com/Niliix007/betree_storage_stack)) implements a key-value store and is written in Rust. It can be used directly on top of block storage (that is, an HDD or an SSD) and supports copy-on-write, checksums, snapshots and other convenient features. The underlying data structures use B-epsilon trees, which allow limiting fragmentation by making use of larger node sizes.

We would like to update and extend the implementation to improve its performance and robustness. Moreover, it should offer an object store interface in addition to the key-value interface. This topic requires the storage stack to be extended in Rust and, optionally, a JULEA backend to be written in C. Depending on personal preference, the underlying tree structures can also be modified and optimized.

More background information can be found in a [previous thesis](https://wr.informatik.uni-hamburg.de/_media/research/theses/felix_wiedemann_modern_storage_stack_with_key_value_store_interface_and_snapshots_based_on_copy_on_write_b%CE%B5_trees.pdf) ([https://wr.informatik.uni-hamburg.de/\\_media/research/theses/felix\\_wiedemann\\_modern\\_storage\\_stack\\_with\\_key\\_value\\_store\\_interface\\_and\\_snapshots\\_based\\_on\\_copy\\_on\\_write\\_b%CE%B5\\_trees.pdf](https://wr.informatik.uni-hamburg.de/_media/research/theses/felix_wiedemann_modern_storage_stack_with_key_value_store_interface_and_snapshots_based_on_copy_on_write_b%CE%B5_trees.pdf)).

Contact: [Michael Kuhn](https://parcio.ovgu.de/People/Michael+Kuhn.html) (<https://parcio.ovgu.de/People/Michael+Kuhn.html>)