
Design and Implementation of an Object Store with Tiered Storage

- ▶ Author: Till Höppner
- ▶ Type: Bachelor's Thesis
- ▶ Date: 2021-08-12
- ▶ Reviewers: Jun.-Prof. Dr. Michael Kuhn, Kira Duwe
- ▶ Supervisors: Jun.-Prof. Dr. Michael Kuhn
- ▶ Download: PDF

The B-epsilon-tree storage stack is a library built upon the B-epsilon-tree data structure, which manages block devices directly to offer a key-value interface to embedding Rust and C applications. In order to exceed its size limitations, a higher-level object abstraction is built on top of the key-value store by means of splitting into fixed-size chunks. It offers a convenient and misuse-resistant interface, as well as support for user-provided metadata. In addition to a command-line management and analysis tool, the object store functionality is integrated into the JULEA storage framework as a new object backend.

Given the varying performance properties and costs of modern storage devices, an advanced system has the potential of partially realising the benefits of more expensive homogeneous storage by combining different types and utilising each to their strengths. Two approaches of combining different storage devices into a tiered storage system are explored and implemented. Under the assumption that the user is aware of an object's underlying structure, the selected approach allows for a user-specified prioritisation of object parts. The evaluation results suggest that correct prioritisation can lead to performance gains when the access patterns conform to the object prioritisation, and incurs negligible performance cost when operated with a homogeneous storage pool.
