Paper Review: *“Tucana: Design and Implementation of a Fast and Efficient Scale-up Key-value Store”*

Seri Lee
Seoul National University
sally20921@snu.ac.kr

***Abstract*—Key-value stores are the fundamental components for storing and retrieving data. It is essential to persist data even in the case of a power failure, which could happen at any time. Crash-safe key-value stores are difficult to implement. They maintain large, complex, on-disk data structures that atomically update.**

***Index Terms*—Copy-on-Write, Persistence, Crash Recovery, B-Tree**

I. Introduction

Persistent key-value stores are designed to survive machine failures without losing data. It is difficult to build services that can gracefully handle failure at any time, while still providing good performance.

II. Copy-on-Write B-trees

Many file systems use copy-on-write (COW) betrees to provide atomicity, crash recovery, and efficient clones. When the tree would modify a node, it copies the node to a new location on disk, then modifies the copy. Then the parent must be updated to point to the new location. Changes propagate up to the root of the tree. When operations completes successfully, the old root can be discarded. If the system crashes, copy-on-write preserves the original tree. Creating multiple cones is easy and efficient with COW, since trees share as many unmodified blocks as possible. The b-tree reserves the first block of the disk as a superblock. This block contains the block numbers of the root nodes of the top-level b-tree and each allocator tree. Once an operation completes successfully, the superblock is atomically updated with the new roots. If the system crashes before updating the superblock, the tree remain. Mark dirty is an important optimization for COW b-trees.

III. Conclusion

References

[1]
[2]