ELECT: Enabling Erasure Coding Tiering for LSM-tree-based Storage

Yanjing Ren[1], Yuanming Ren[1], Xiaolu Li[2], Yuchong Hu[2], Jingwei Li[3], Patrick P. C. Lee[1]

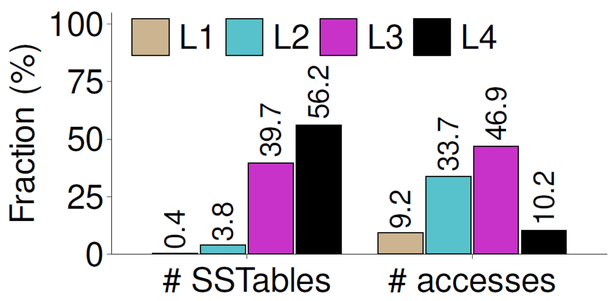
The Chinese University of Hong Kong[1], Huazhong University of Science and Technology[2]

University of Electronic Science and Technology of China[3]

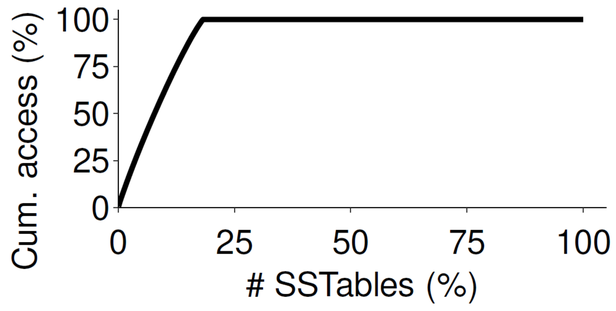
Source code: https://github.com/adslabcuhk/elect

Motivation

Problem: Replication incurs high storage overhead in distributed key-value stores

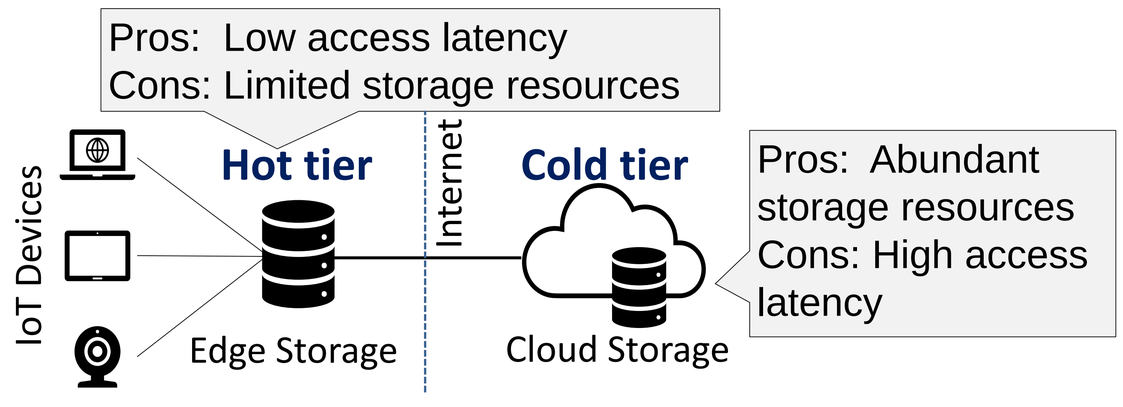


(a) Statistics across levels



(b) Access distributions in L_4

Storage and access patterns in Cassandra



Edge-cloud storage: Example of storage tiering

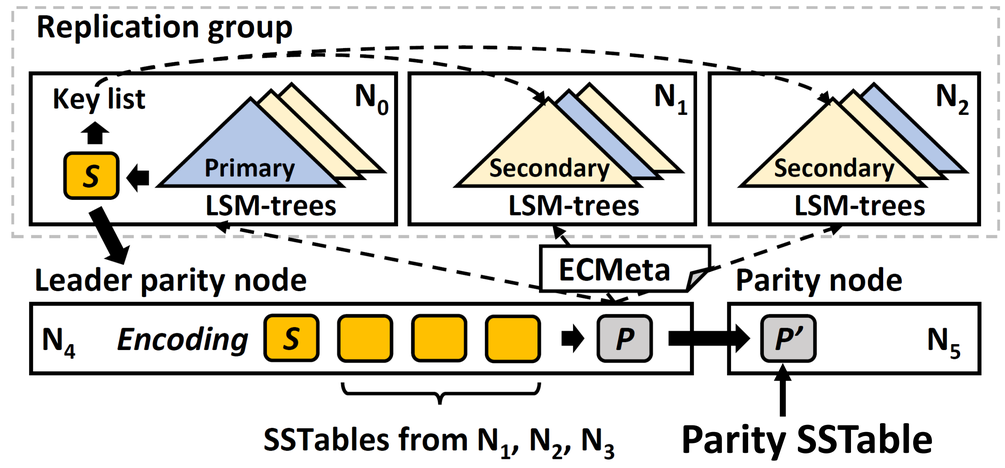
- **Erasure coding** provides low redundancy fault-tolerance for distributed storage
- Skewed workloads are common in practical key-value stores
- We explore **storage tiering** for LSM-tree-based storage to separate data into hot and cold tiers

Main Idea

- **ELECT**, a distributed KV store that enables erasure coding tiering
  - Extends LSM-tree with **hybrid redundancy** by storing hot KV pairs with replication and cold KV pairs with erasure-coding in the hot tier
  - **Offloads cold KV pairs to the cold tier** to further alleviate hot-tier storage overhead

ELECT Design

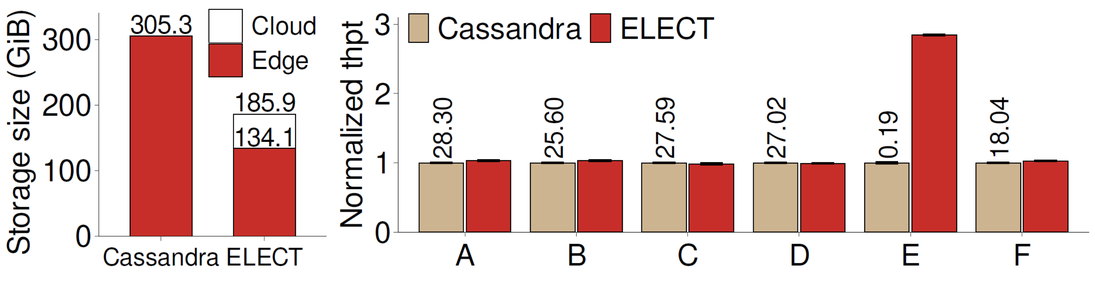
- **Redundancy transitioning**
  - Decouples replicas into multiple LSM-trees
  - Offline cross-encoding of SSTables in last LSM-tree level
  - Load-balanced decentralized parity node selection
  - Fine-grained replica removal
- **Hotness awareness**
  - Monitor SSTables' hotness by access frequency and lifetime
  - Offloading cold data to the cold tier
- **Balancing storage-performance trade-off**
  - Determine SSTables involved in redundancy transitioning and cold-data offloading based on user-specified storage saving target



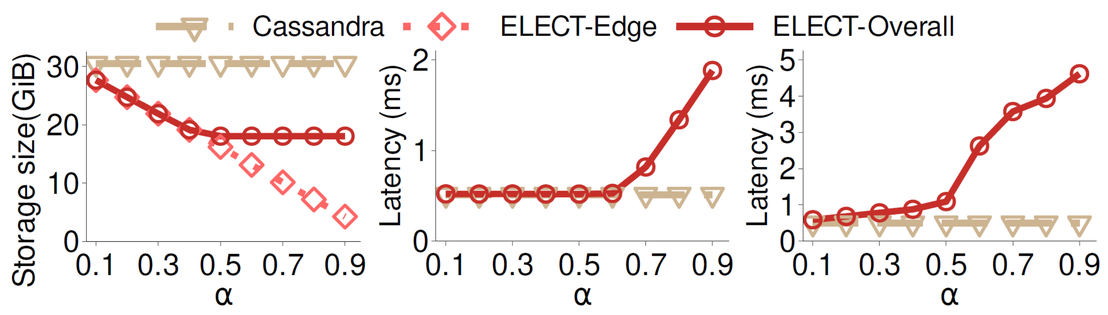
ELECT is prototyped atop Cassandra 4.10.0 (all artifact badges are awarded)

Prototype Experiments

Consider an edge-cloud setting



(a) Storage size (b) Throughput



(a) Storage size (b) Reads in normal mode (c) Reads in degraded mode

Performance of YCSB core workloads

- *Storage overhead:* **56.1%** edge storage saving from Cassandra
- *Performance:* Preserve Cassandra throughput (up to **3%** difference except for workload E)

Performance with different storage saving targets α

- ELECT reduces edge storage overhead by **9.2-86%** over Cassandra (4% difference from α)
- ELECT maintains read latency in normal mode with a lower storage saving target (i.e., $\alpha \leq 0.6$)