**The Problem:** LSM-Trees Have Suboptimal Read Performance

- Recently updated keys are the cheapest to access.
- Likely to be read next are not necessarily the most likely to be read next.

**The Solution:** Splaying

- Copy frequently accessed keys to top.
- Use ‘get’ as a trigger for possible ‘put’.
- ...and over time the tree is reorganized for better read performance.

**Compared with the state-of-the-art** solution of adding a cache:

- Splaying dominates cache in the absence of Bloom Filters.
- ...and can offer adaptivity and better performance across workloads.