Compaction management in distributed key-value datastores

Muhammad Yousuf Ahmad
muhammad.ahmad2@mail.mcgill.ca

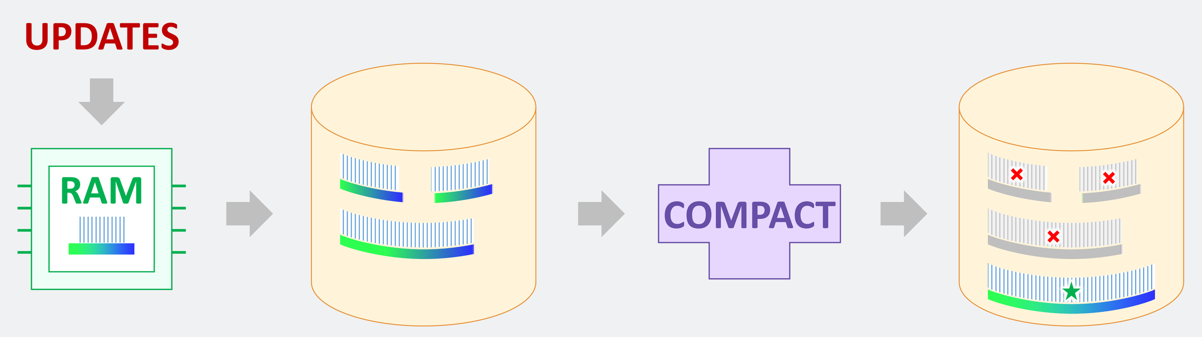
Bettina Kemme
kemme@cs.mcgill.ca

McGill UNIVERSITY

COMPACTIONS

Periodic data maintenance for multi-version stores

– Consolidate updates into existing dataset



Prevent read latency from degrading over time

– Reduce number of overlapping data files to be read

Compactions are expensive!

1. **During** execution: compete with workload for resources
2. **After** execution: degrade read latency severely
   – Input files removed, evicted en masse from cache
     ▶ Cache misses

GOALS 1. Reduce compaction overheads on region server.
2. Prevent large spikes in read latency.

SOLUTION

≡ **Compaction Offloading**

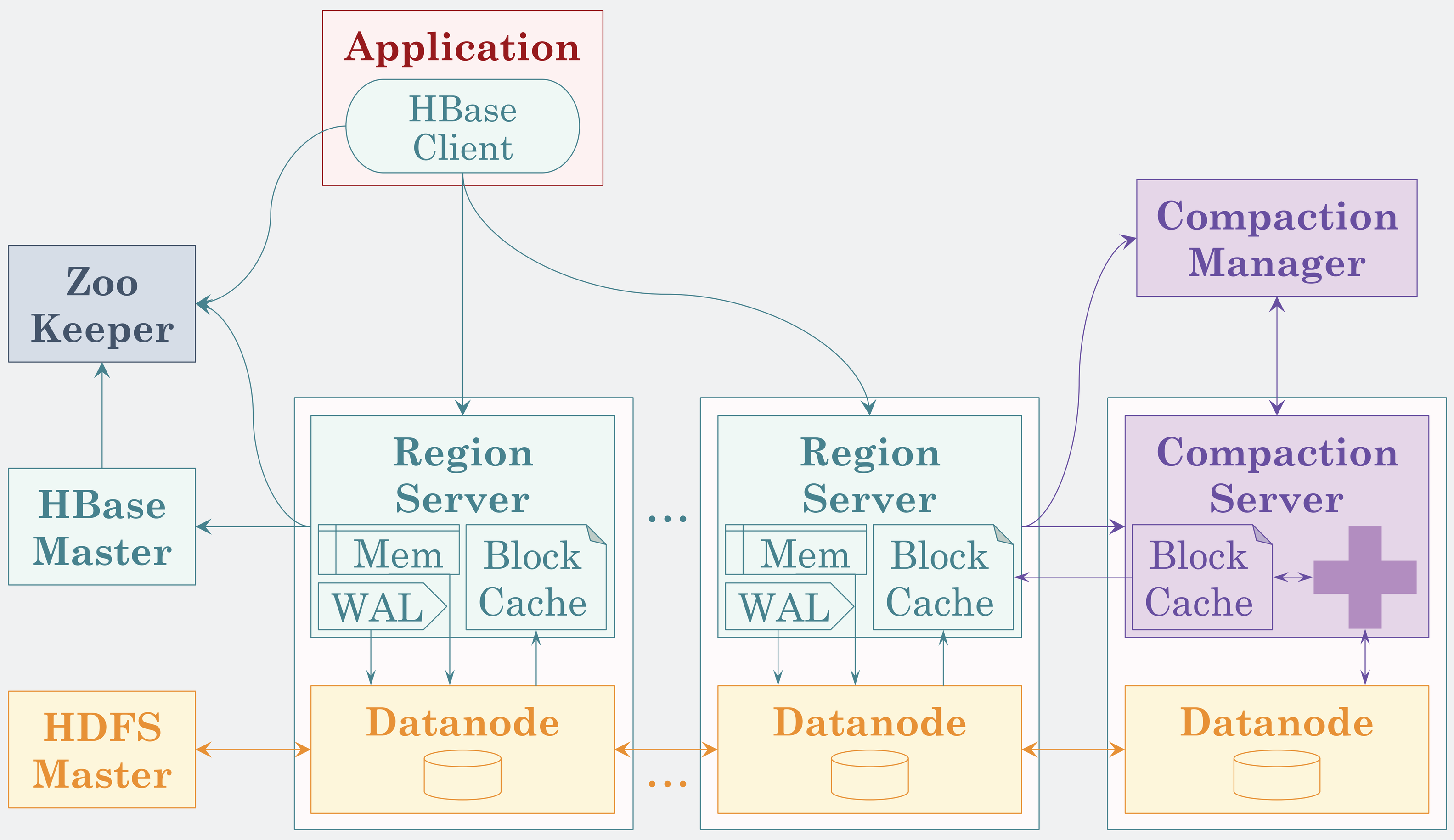
– Offload compactions to specialized ***compaction servers***.
– Dedicate region server resources to workload execution.

≡ **Remote Caching**

– Compaction server caches compaction results locally.
– Region server reads back results over network.

≡ **Incremental Warmup**

– Do *not* evict invalidated file blocks en masse.
– Gradually phase old data out, block by block.
– Replace with new data from remote cache.
– Sequential transfer (files are already sorted).



RESULTS

– Compaction server assumes execution and overheads.
  – Compactions are shorter; read latency improves.
– Cache misses less costly; reads faster over network vs. disk.
– Incremental warmup eliminates cache misses altogether.
– Multiple compaction servers allow for load balancing.

EXPERIMENTS

– YCSB update workload triggers compactions.
– YCSB read workload measures get/scan latency.
– 1x region server (RS), 1x compaction server (CS).

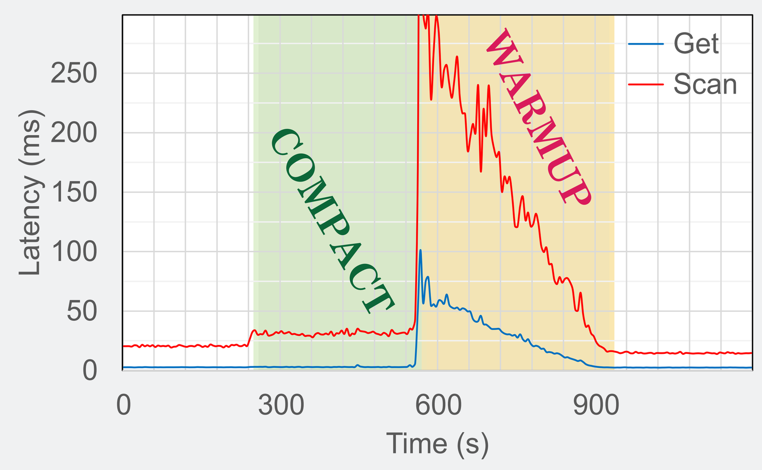


Fig. 1 – Standard Compaction

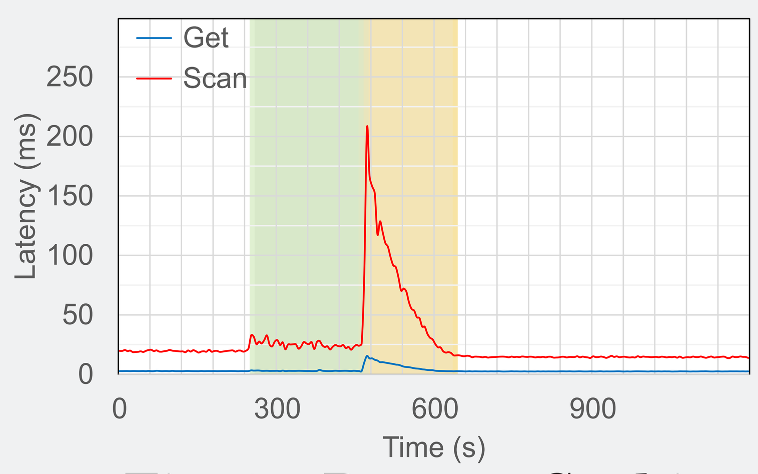


Fig. 2 – Remote Caching

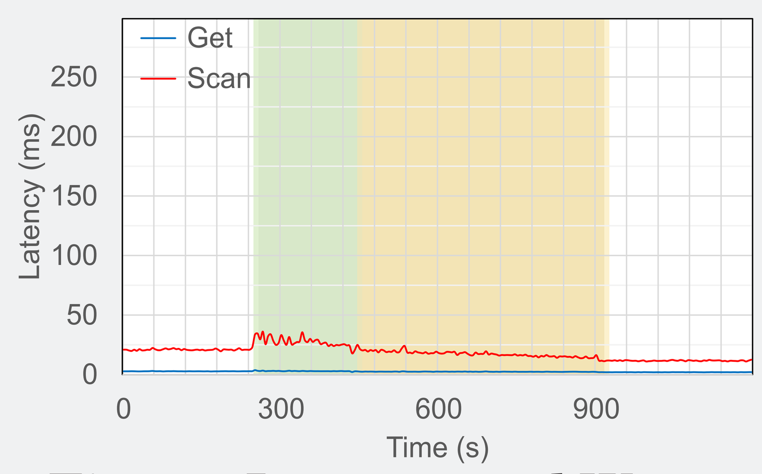


Fig. 3 – Incremental Warmup

SCALABILITY

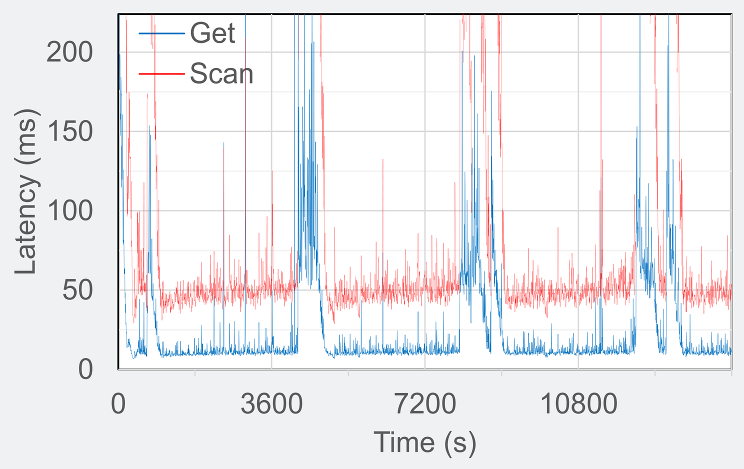


Fig. 4 – Standard
5x RS - No CS

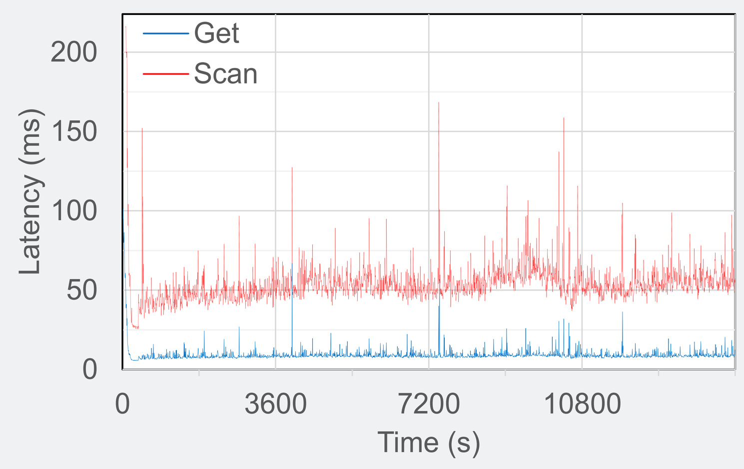


Fig. 5 – Offloaded
5x RS - 1x CS

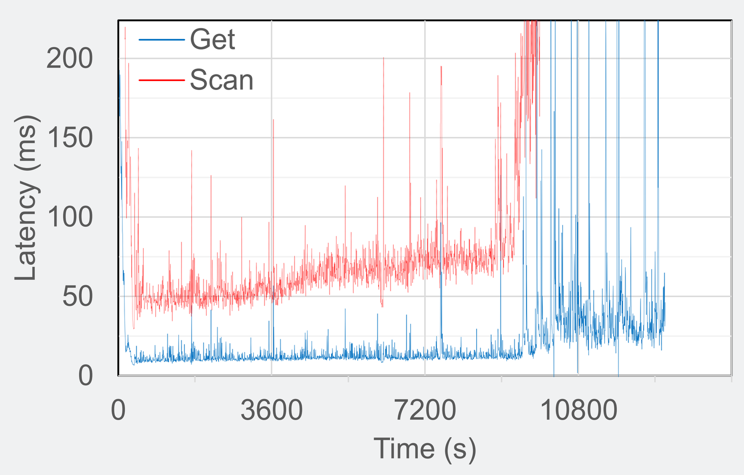


Fig. 6 – Under-Provisioned
10x RS - 1x CS

Compactions pile up, overloading the single compaction server.

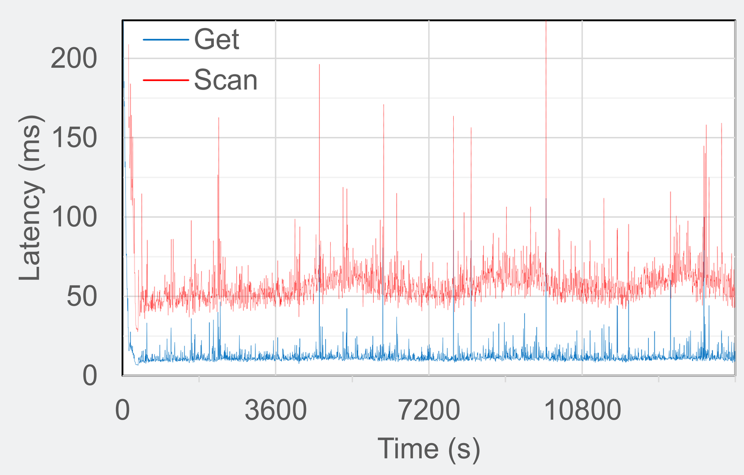


Fig. 7 – Balanced
10x RS - 2x CS

Compactions are distributed across two compaction servers.

REFERENCES

[1] MY Ahmad and B Kemme. Compaction Management in Distributed Key-Value Datastores. *PVLDB* 8(8):850–861, 2015.

[2] AS Aiyer, M Bautin, GJ Chen, P Damania, P Khemani, K Muthukkaruppan, K Ranganathan, N Spiegelberg, L Tang, and M Vaidya. Storage infrastructure behind Facebook Messages: Using HBase at scale. *IEEE Data Eng. Bull.*, 35(2):4–13, 2012.

[3] F Chang, J Dean, S Ghemawat, WC Hsieh, DA Wallach, M Burrows, T Chandra, A Fikes, and R Gruber. Bigtable: A distributed storage system for structured data. In *OSDI*, pages 205–218, 2006.

[4] PE O'Neil, E Cheng, D Gawlick, and EJ O'Neil. The log-structured merge-tree (LSM-tree). *Acta Inf.*, 33(4):351–385, 1996.

[5] Apache HBase. http://hbase.apache.org/

...

ACKNOWLEDGEMENTS

This work is partially funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) and Ministère de l'Enseignement supérieur, Recherche, Science et Technologie, Québec, Canada.