A Contextual Bandit for Dynamic Data Movement in Detock

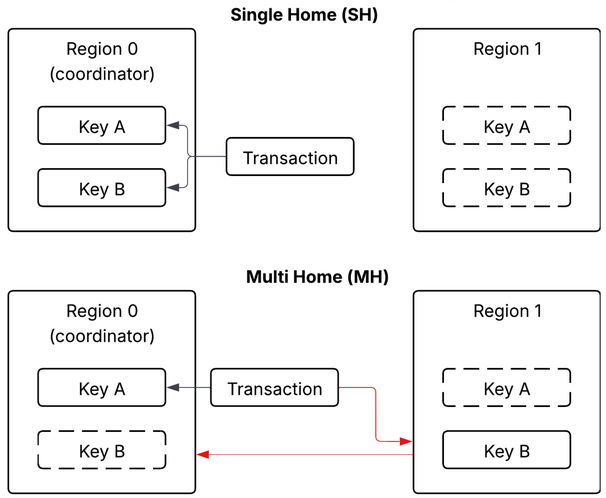
Rares Andrei Popa | Supervisors: Asterios Katsifodimos, Oto Mráz | EEMCS, Delft University of Technology

TU Delft

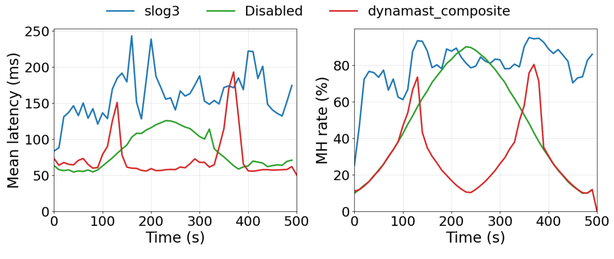
Problem

Records have a *home region*. Remastering changes a record's home — but it causes concurrent transactions to restart.

Detock has the mechanism, but no policy.



Motivation



- *SLOG*: per-key remaster splits co-accessed groups.
- *DynaMast-style*: spikes at transitions.

Placement must adapt. Co-accessed keys must move together.

Approach

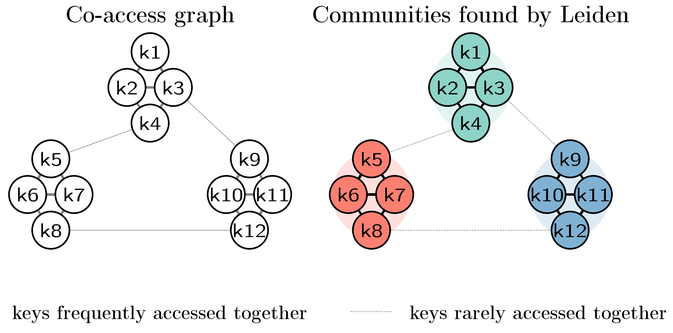
Each round:

1. Build co-access graph over *hot* keys
2. Leiden clustering → migration communities
3. Bandit picks action per community
4. Execute; delayed reward trains the network

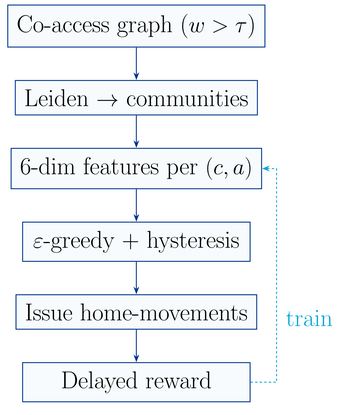
Edge weight:

$$w(k_i, k_j) = \frac{\text{co-occurrences}(k_i, k_j)}{\min\big(\text{count}(k_i),\ \text{count}(k_j)\big)}$$

Leiden clustering:



Pipeline:

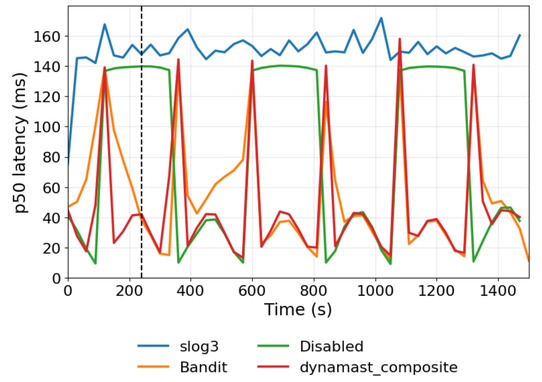


Reward

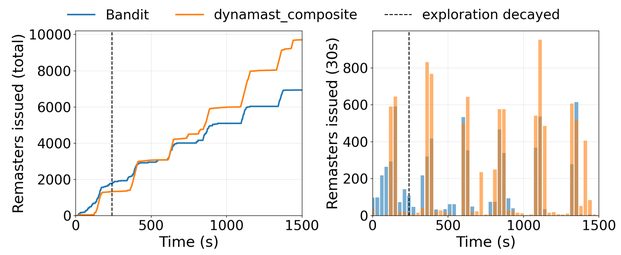
$$r = 2 \cdot \max\big(-1,\ (\text{local\_frac} - \text{cf\_frac}) - \text{restart\_frac}\big)$$

- **local_frac**: accesses served by the chosen master.
- **cf_frac**: counterfactual locality of best alternative.
- **restart_frac**: restarts caused by this decision.

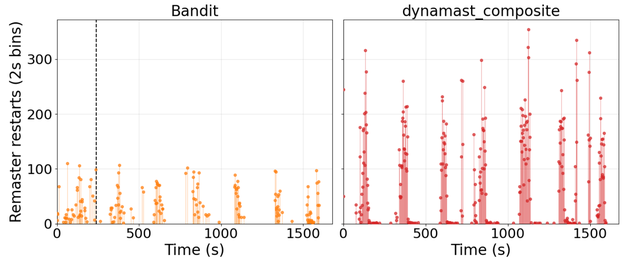
Results



Median latency. SLOG never converges. Bandit and DynaMast-style both track the hotspot. The difference is in *cost*.



Migrations issued. Bandit bursts at rotation boundaries, idles between. DynaMast-style migrates continuously — 29% fewer restarts.



Restarts. DynaMast-style migrates many keys in multiple episodes, widening the stale-home window — 4.3× more restarts.

Conclusions

- **Community migration** prevents co-access splitting.
- **Restart penalty** in the reward prevents wasteful moves.
- **No prior knowledge** — learns from runtime only.

Future work. Warm-starting; distance-aware reward; incremental communities.