Control-Based Dynamic Data Movement for Geo-Distributed Systems

Final Research Poster

Arpad Jakab Oto Mraz[1]
Delft University of Technology
[1]Supervisor

CSE3000 Research Project, June 2026

Problem & Research Question

- Detock assigns every data item a **home region**.
- When requests originate near that home, transactions can be handled locally. Requests from another region require **remote forwarding**; transactions spanning multiple homes require additional cross-region coordination.
- Static placement can become inefficient when workload locality changes over time.
- **Remastering** can move a key's home to better match current demand, but unnecessary movement may introduce overhead, aborts, and ownership churn.

Research question. How does a control-based policy for dynamic data movement affect locality, latency, throughput, and movement overhead in Detock under changing workload locality?

Remastering in Detock

Goal: align a key's home with the region that persistently requests it.

Before		After
Home: **Region 0** Demand: **Region 1** *Remote forwarding*	⇒ Remaster key *x*	Home: **Region 1** Demand: **Region 1** *More local requests*

Only keys with sufficient recent evidence are moved; the cooldown reduces immediate reversals and movement limits bound the amount of work issued at once.

Sliding-Window Policy

For every access, the controller records the requesting region in a bounded history for that key.

1. **Observe access** Key *x* requested from Region *r*
2. **Update recent history** Keep the latest 50 requester-region observations for *x*
3. **Evaluate dominant region** Every 500 ms, find which region dominates the window
4. **Remaster only if stable** Dominant share ≥ 0.80, enough samples, different home, cooldown passed

Final policy configuration

Window = 50 accesses
Threshold = 0.80
Minimum samples = 20
Cooldown = 20 accesses
At most 4 remasters per epoch
At most 3 in flight

Threshold and window size were selected using one-trial screening followed by repeated validation.

Experimental Setting

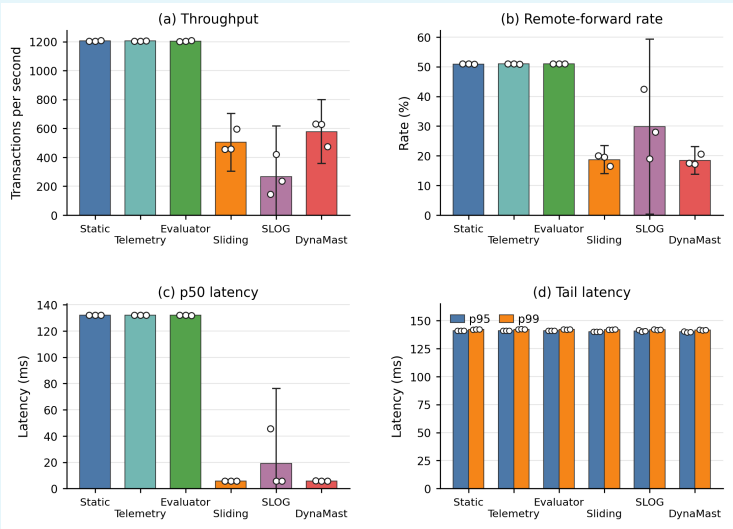
System. Detock deployed on four cluster nodes across two logical regions, with two partitions per region.

Network. Intra-region RTT was approximately 0.15 ms; inter-region RTT was set to **100 ms**, modelling a US West–EU West deployment.

Primary workload. Controlled YCSB locality-shift scenario with 50 clients per region. At $t = 30$ s, the dominant requester region changes for selected policy-test keys while the total client population and load remain fixed.

Protocol. 90-second runs with a 10-second warmup; three trials per configuration. Figures report means, individual trials, and 95% Student-*t* confidence intervals.

Primary Results



Means over three trials. Dots show individual trials; error bars show 95% Student-t confidence intervals.

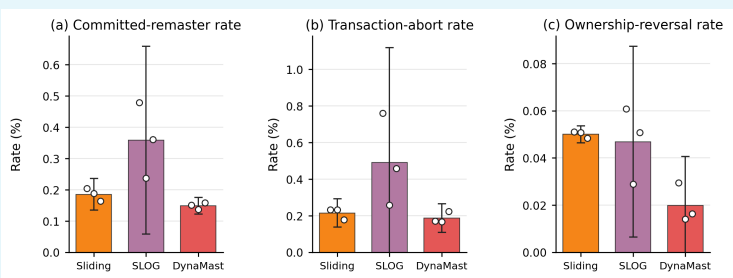
Sliding window versus static placement

- Remote forwarding falls from **50.9%** to **18.7%**.
- Median latency falls from about **132 ms** to **5.8 ms**.
- Throughput falls from about **1205** to **504 transactions/s**.

The static baseline and the no-movement controls remain near 1204–1206 transactions/s and 50.9% remote forwarding. Therefore, the substantial throughput reduction appears when remastering is enabled, not from telemetry collection or periodic evaluation alone.

Interpretation. Remastering makes enough requests local to transform p50 latency, but the remaining remote or transition-heavy transactions keep p95/p99 near the WAN-latency regime.

Movement and Transition Costs



Rates are percentages of committed transactions.

For the sliding-window policy, the committed-remaster rate was about **0.19%**, the observed transaction-abort rate was about **0.21%**, and the ownership-reversal rate was about **0.05%**.

Important: the abort bars represent benchmark transactions interrupted during ownership transitions. They are not failed remaster operations; remaster-operation aborts were zero in the valid final trials.

Parameter Selection

Threshold: 0.80 Initial one-trial screening tested thresholds 0.60, 0.70, 0.80, and 0.90. Repeated validation then compared 0.70 and 0.80. Throughput and latency overlap substantially; 0.70 has slightly lower remote forwarding, while 0.80 has lower remaster and reversal rates. Thus, 0.80 was selected as the more conservative operating point.

Window: 50 accesses Initial one-trial screening tested windows 50, 100, 200, and 400 at threshold 0.80. Larger windows retained more pre-shift observations and adapted more slowly. Repeated validation then compared windows 20, 30, 40, and 50. Window 50 had the highest observed mean throughput and lowest observed remote-forward rate; p50 and p95 were similar across windows.

Additional safeguards

- At least 20 observations before evaluation.
- 20-access cooldown after a remaster.
- At most 4 remasters per evaluation epoch.
- At most 3 remasters in flight.

Threshold and window size were selected using one-trial screening followed by repeated validation.

Conclusion & Scope

Conclusion. Under controlled locality shifts, the sliding-window policy successfully adapts key ownership. Remote forwarding falls from **50.9%** to **18.7%**, and p50 latency falls from about **132 ms** to **5.8 ms**.

However, throughput falls from about **1205** to **504 transactions/s**. The current implementation improves locality and median latency, but does not demonstrate an overall performance improvement in this setting because aggregate throughput falls.

Scope

- Controlled two-region YCSB locality-shift workload.
- Three trials per configuration; movement outcomes remain variable.
- Per-key decisions do not account for groups of keys accessed together.

Future work. Estimate expected locality benefit before moving a key, model co-accessed key groups, and reduce the overhead of ownership transitions.

TU Delft Delft University of Technology