Virtual Consensus in Delos

Balakrishnan et al., OSDI ‘2020

Consensus-based database replication

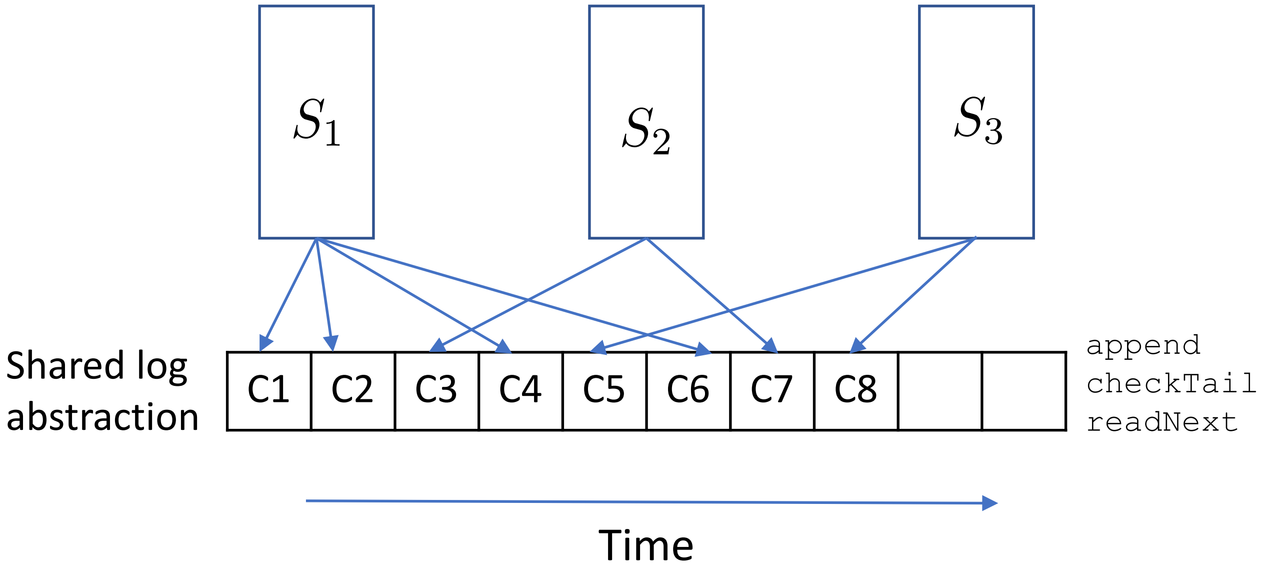
Shared log abstraction:

- append
- checkTail
- readNext

achieved by

Fault-tolerant consensus:

- Ordering protocol
- Leader election
- Membership changes



Problem

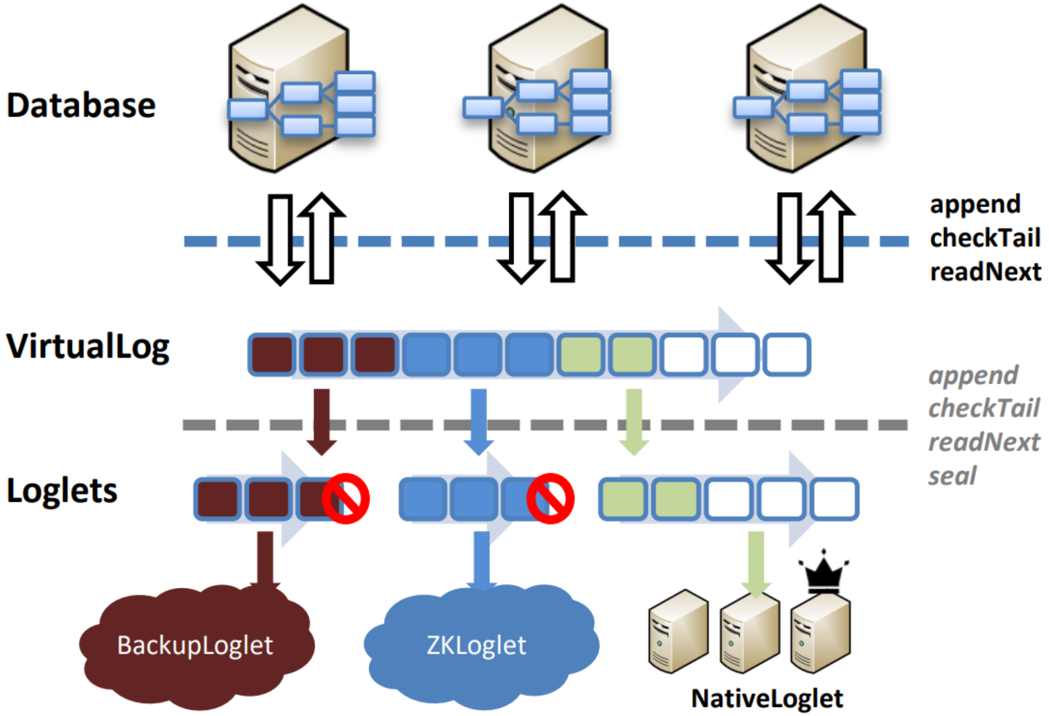
- Consensus-based database replication systems are monolithic, complex, and difficult to evolve.
- Shared log a good abstraction, but implementations are:
  - *Difficult to deploy and operate:* no support for upgrading and migrating applications without downtime
  - *Difficult to develop:* monolithic consensus protocols with coupled control plane and data plane

Fault-tolerant consensus

[Data plane] Ordering protocol
[Control plane] Leader election
[Control plane] Membership changes

Virtual consensus

- Layered approach:
  - VirtualLog for *control plane* (reconfiguration, leader election, membership changes)
  - Loglets for *data plane* (ordering commands and storing them durably)



Key benefits

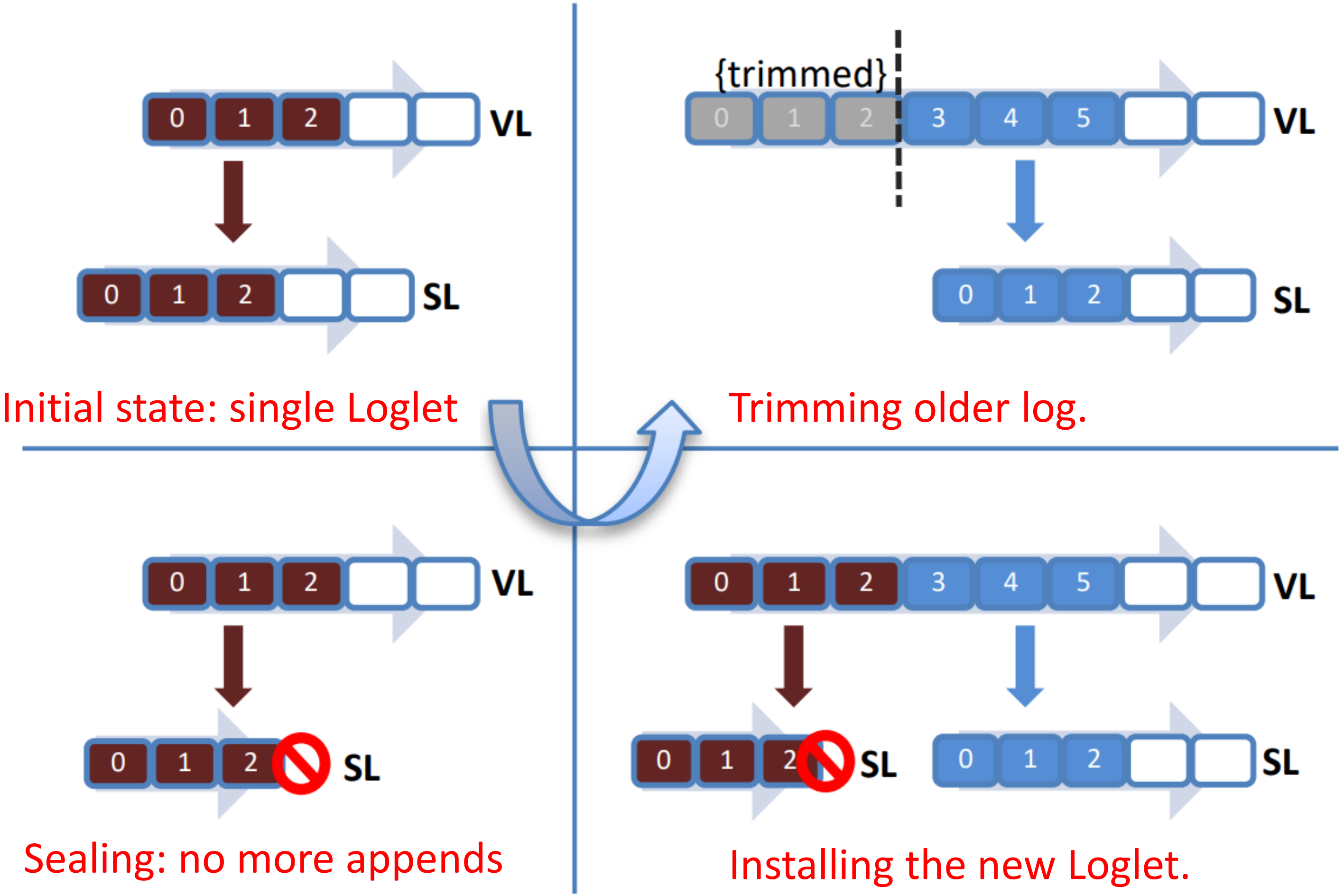
- Applications agnostic of VirtualLog’s virtual nature, and see the traditional shared log API.
- Loglets do not need to implement a full fault-tolerant consensus protocol.
    - Can only implement a simple ordering protocol.*
- VirtualLog’s fault tolerant consensus does not need to be efficient, as it is triggered only during reconfiguration.
- Allows hot-swapping of Loglets as scalability requirements change.

*: Loglets only need to implement a fault-tolerant `seal’ operation, which is theoretically weaker and much simpler to implement

Virtual consensus in Delos

- Delos: A database running on top of the VirtualLog API.
- *Simplified deployment and operation:*
  - Rapid initial development with rudimentary Loglet implementation
  - 10X improvement in latency by hot-swapping the Loglet implementation in production to a more efficient one.
  - Migrating older segments to a Loglet layered on cold storage.
- *Simplified development:*
  - *Loglets are simple to design:* Delos with a primary sequencer-based Loglet protocol.
  - *Loglets can be composed:* A StripedLoglet to achieve the following:
    - Double the failure threshold by rotating the sequencer.
    - Support >1M appends/sec by sharding.

VirtualLog reconfiguration



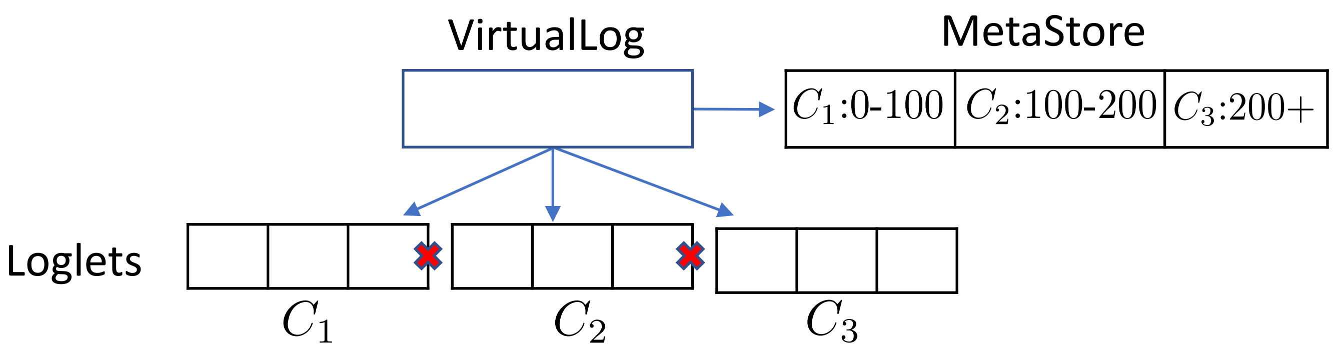
Initial state: single Loglet

Trimming older log.

Sealing: no more appends allowed on this Loglet.

Installing the new Loglet.

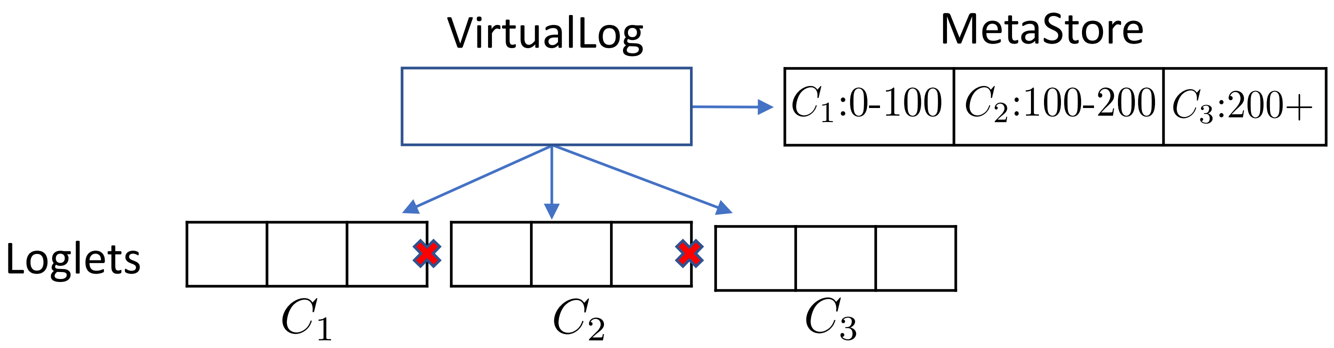
VirtualLog reconfiguration protocol



Versioned register with conditional write: append C_{i+1} only if current chain is C_i

1. Seal the old chain (seals are idempotent; appends disallowed post-sealing)
2. Install new chain on MetaStore (at most one winner in case of races)
3. Fetch new chain from MetaStore (in case someone else won the race)

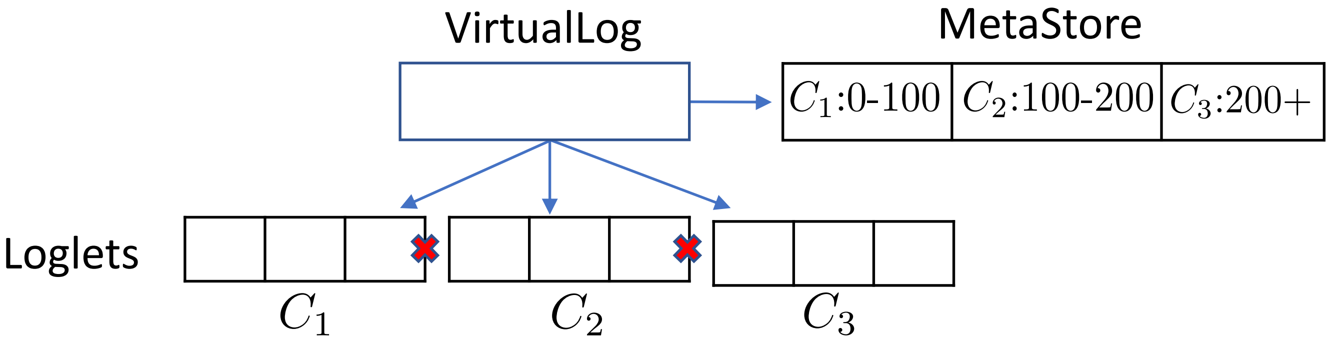
Guarantees



Versioned register with conditional write: append C_{i+1} only if current chain is C_i

- Concurrency:
  - Multiple clients can seal concurrently. Seals are idempotent.
  - `checkTail` to a Loglet should return a *sealed* bit, indicating whether it is sealed or not. If working on a sealed chain, fetch the latest chain from MetaStore and try again.
- Failure atomicity:
  - If no new chain found after sealing, retry after some time and install a new chain with the same configuration as earlier.

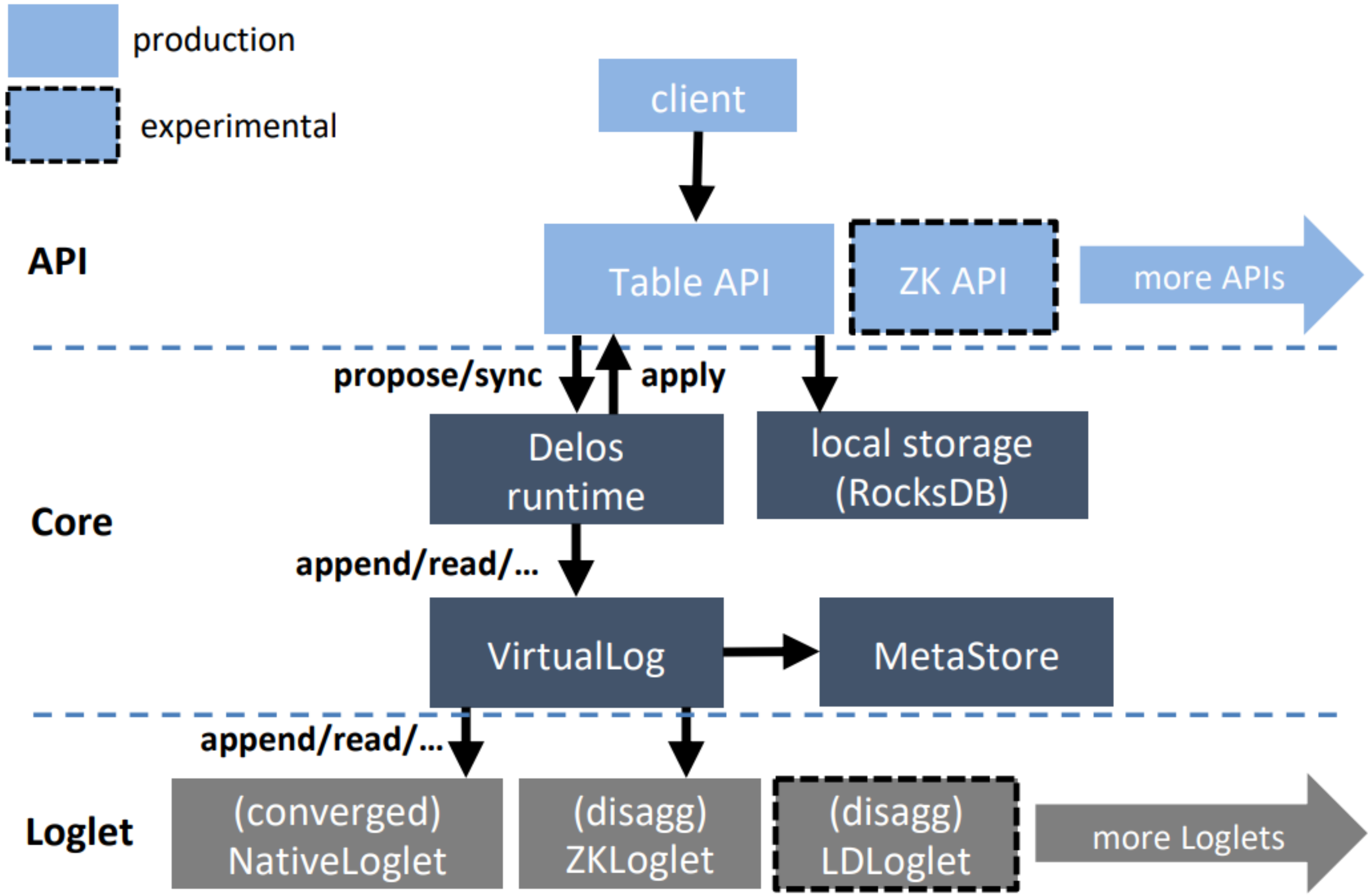
Requirements for components



Versioned register with conditional write: append C_{i+1} only if current chain is C_i

1. MetaStore needs to implement the ‘versioned register with conditional write’, with a fault-tolerant consensus protocol like Paxos.
2. Loglets need to implement the sealing operation in a fault-tolerant way, via a *fault-tolerant atomic register,* which is weaker than consensus.
3. Loglets need to reject any append operations post-sealing.
4. Loglets need to return the *sealed* bit on the checkTail call.
5. Loglets need to detect failures and initiate the VirtualLog reconfiguration process.

Delos – a database server over virtual consensus

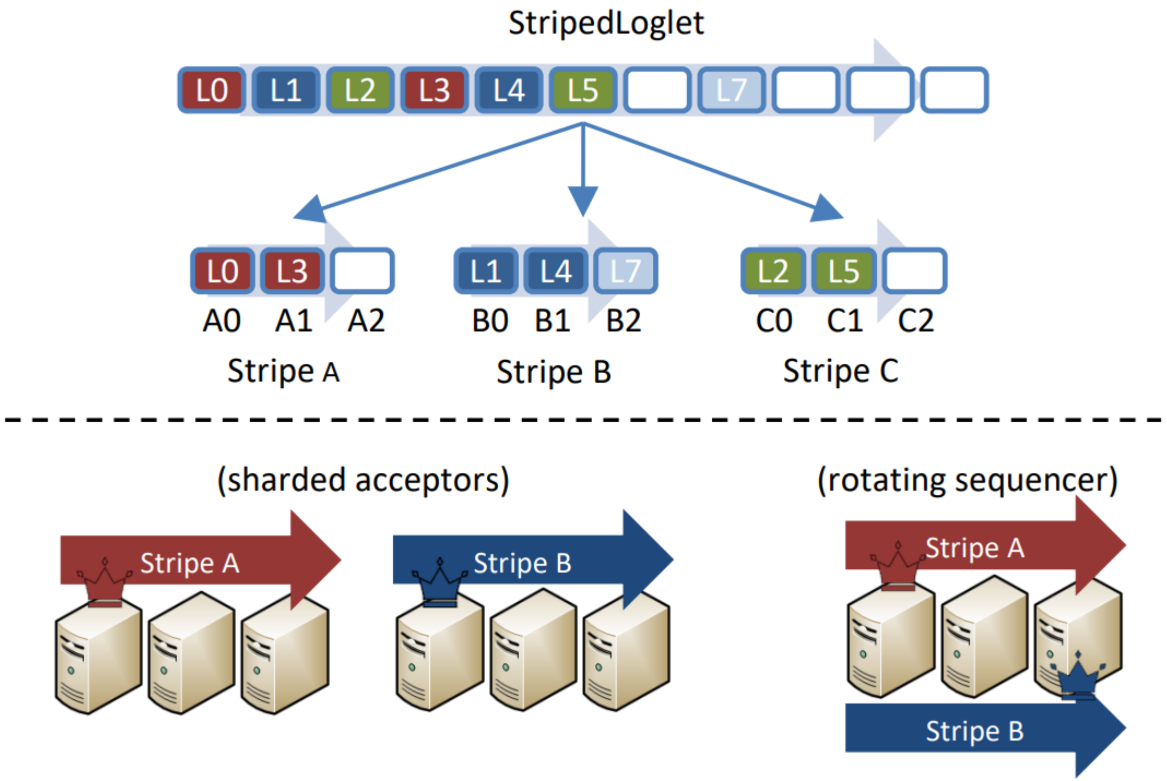


Delos with NativeLogLets

- A primary sequencer-based Loglet protocol:
  - append calls contact a primary sequencer node and fail if the sequencer is down.
  - sealing requests each LogServer to set the seal bit.
  - checkTail implements a protocol to return a sealed status based on the sealed status of each LogServer.

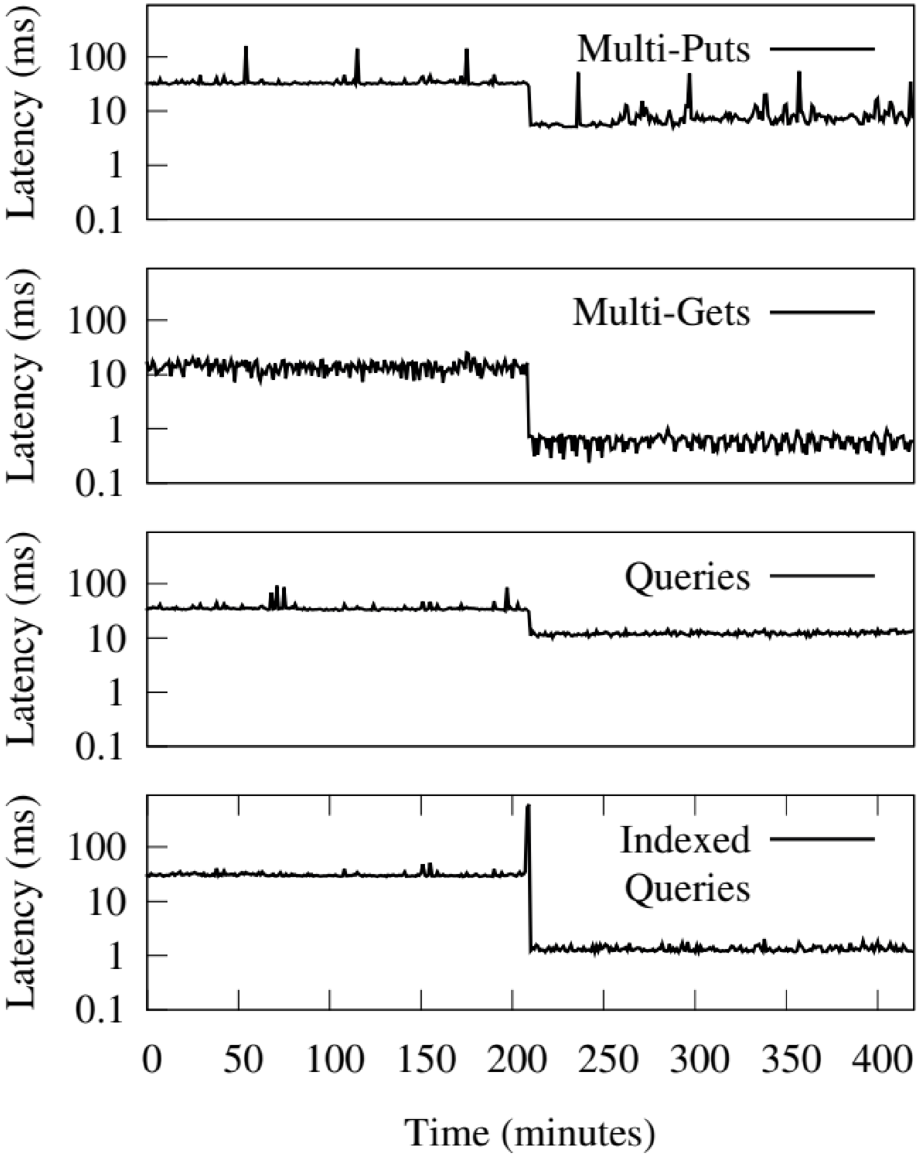
Delos with StripedLogLets

- StripedLoglets compose simple Loglets to get special performance and robustness properties.

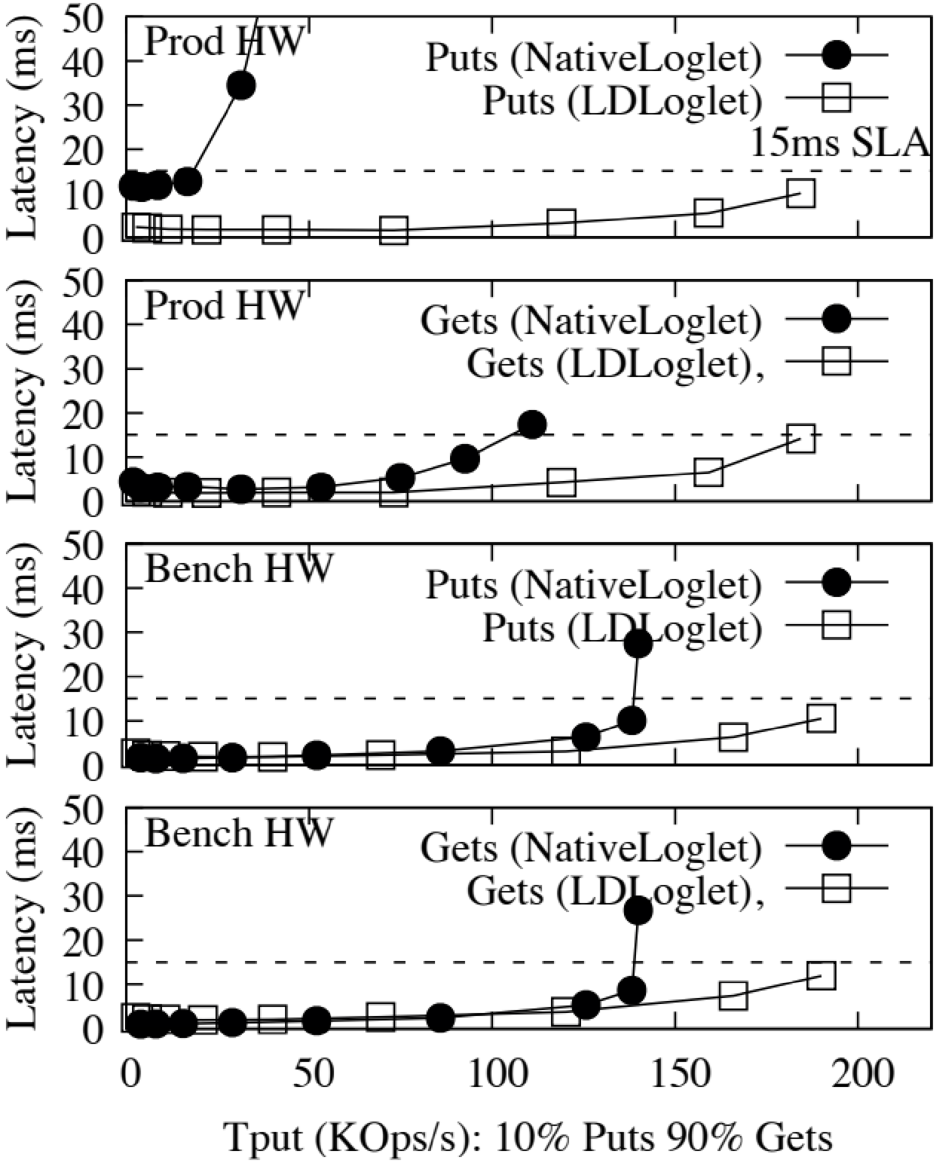


Benchmarks

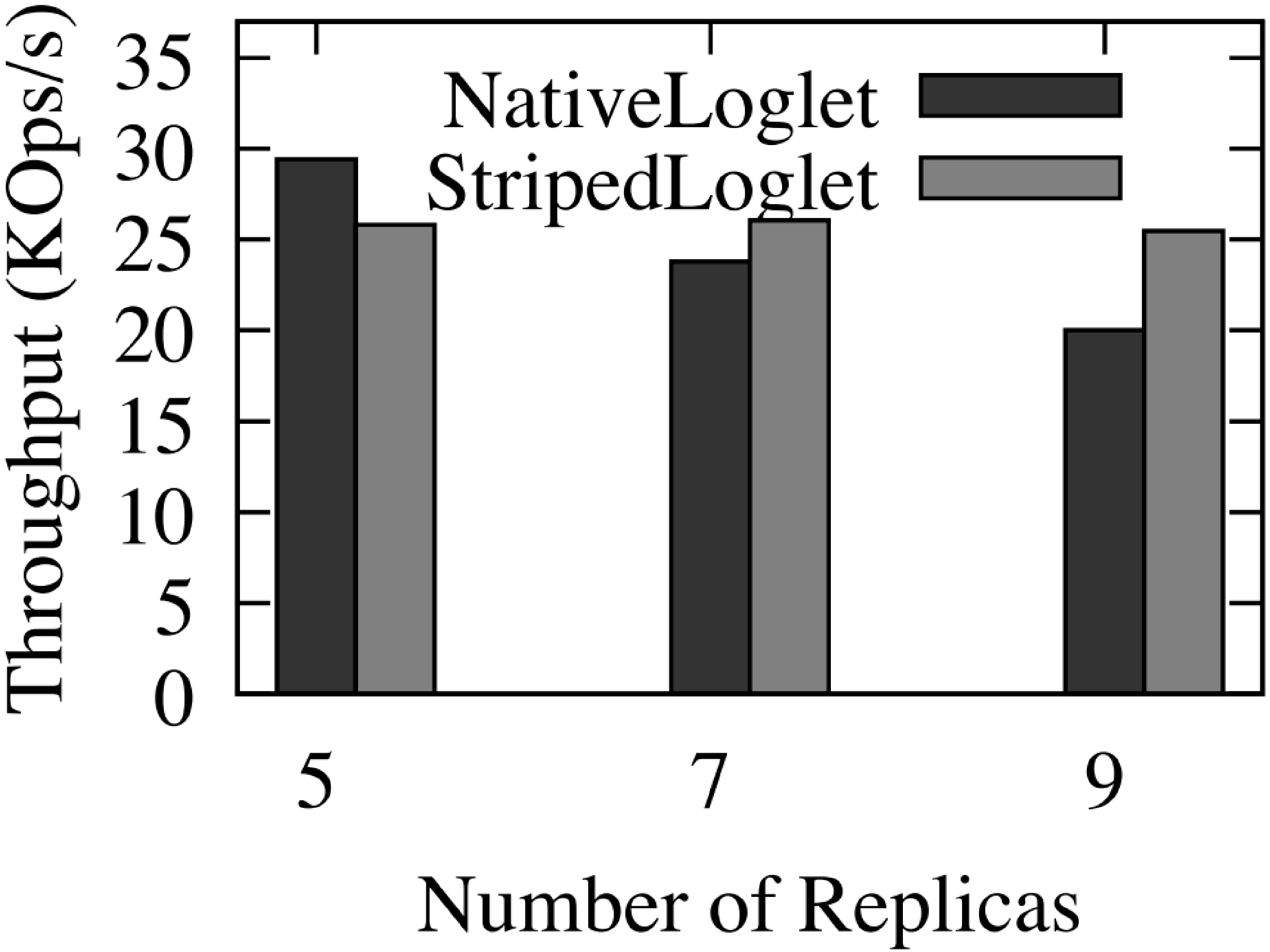
Latency during production switchover of the Loglet protocol



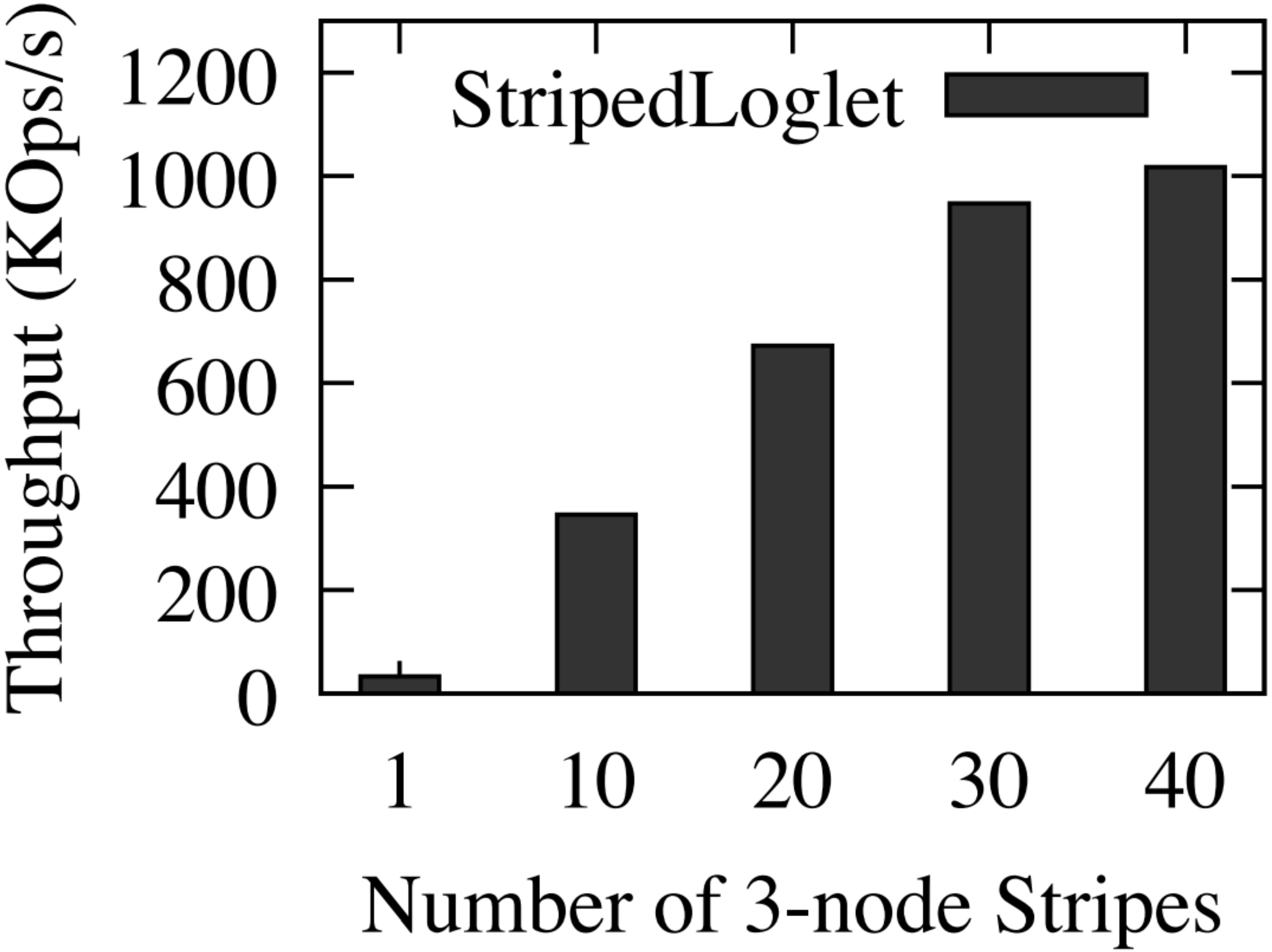
Throughput improvement via disaggregation



Removing sequencer bottlenecks

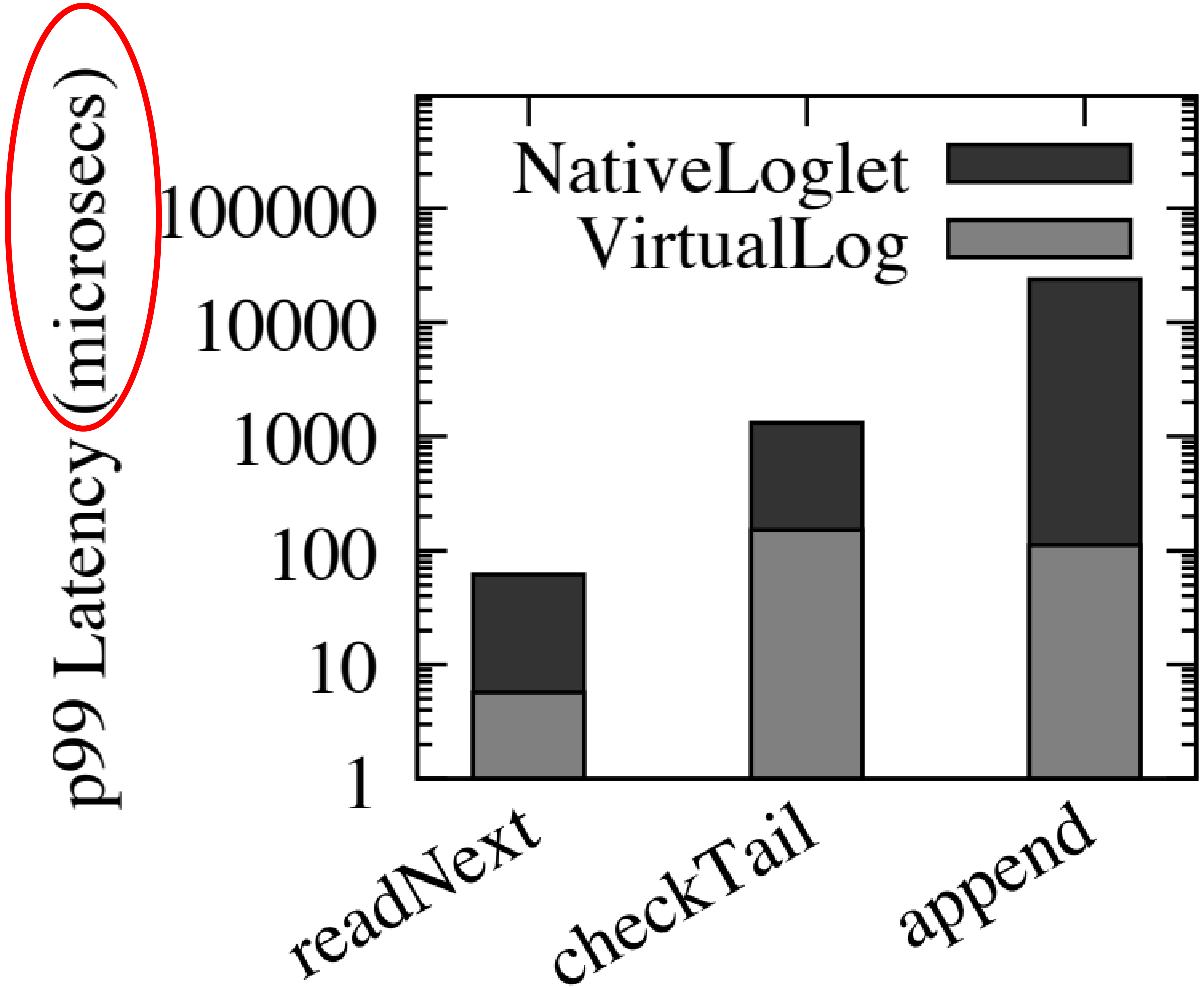


Horizontal scaling



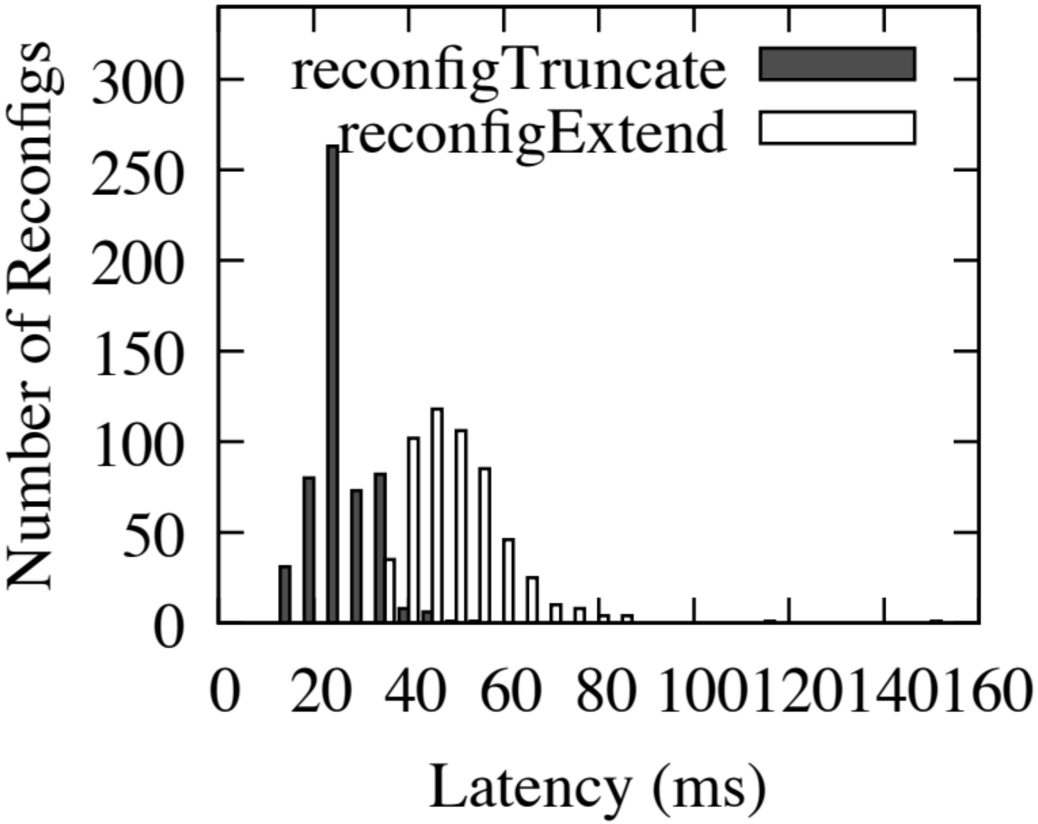
The cost of virtualisation

Virtualisation latency

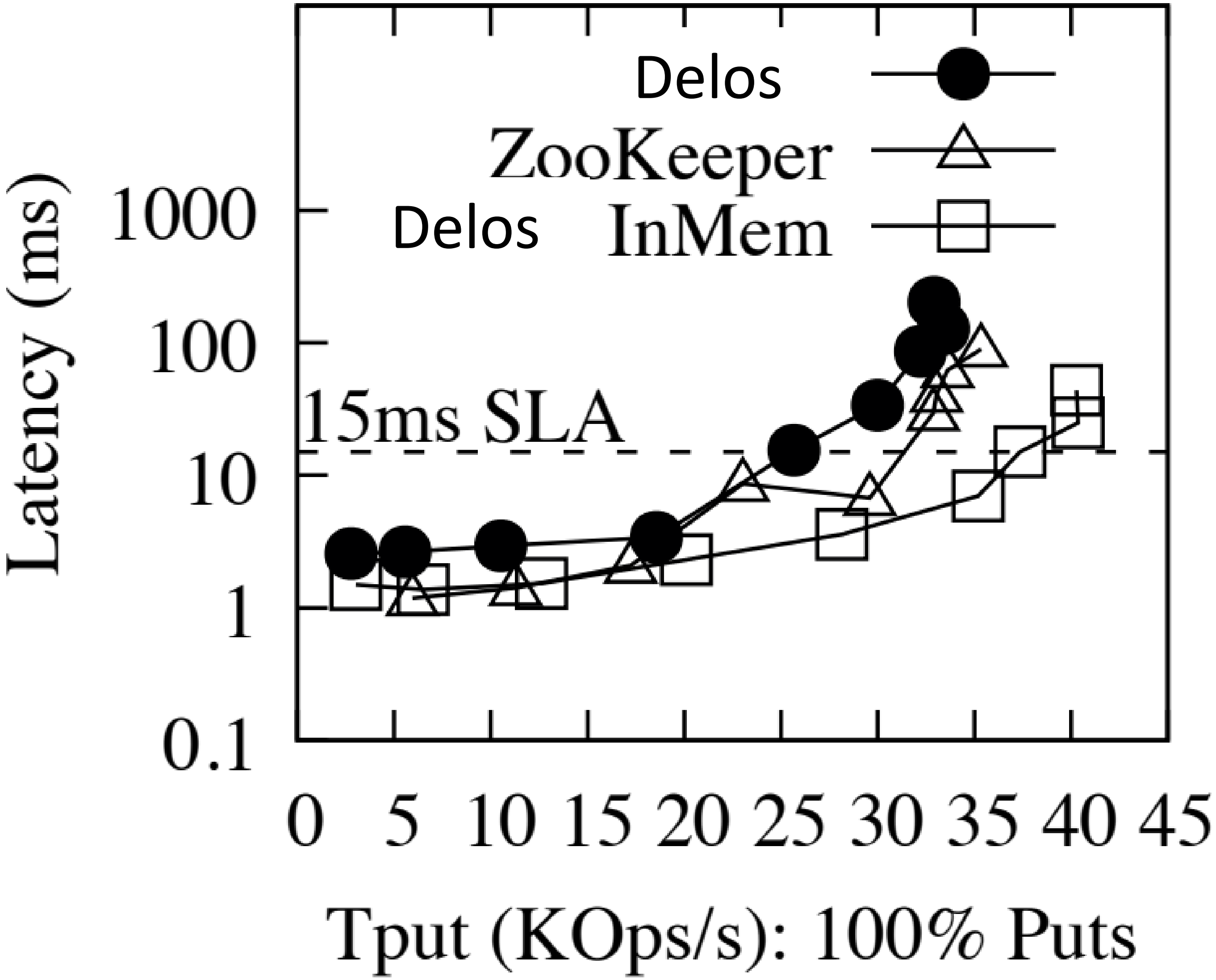


The cost of virtualisation

Reconfiguration latency



Comparison with ZooKeeper


Delos
ZooKeeper
Delos InMem
15ms SLA
Latency (ms)
Tput (KOps/s): 100% Puts