
TigerBeetle
1000X
WORLD TOUR
13 CITIES | 6 DAYS

The Tale of Taming TigerBeetle's Tail Latency

Act I: The Protagonist

TigerBeetle is a high-performance database for financial accounting.

TigerBeetle is a high-performance database for ~~financial ac~~counting.

Debit/Credit Interface

```
Transfer = struct {
    id: u128,
    debit_account_id: u128,  // Who
    credit_account_id: u128, // Who
    amount: u128,            // How much
    timestamp: u64,          // When
    //...                       What, Why, Where
}
```

128 bytes (2 cache lines).

Debit/Credit Interface

```
Transfer = struct {
    id: u128,
    debit_account_id: u128,  // Who
    credit_account_id: u128, // Who
    amount: u128,            // How much
    timestamp: u64,          // When
    //...                       What, Why, Where
}
```

128 bytes (2 cache lines).

Batching, batching, batching.

TigerBeetle's Design Decisions

- strictly serializable
- highly durable and available [1]
  - network and node faults [2]
  - storage fault model[3] [4]
  - deterministic simulation testing and extensive fuzzing

[1] https://jepsen.io/analyses/tigerbeetle-0.16.11
[2] Viewstamped Replication
[3] A Study of SSD Reliability in Large-Scale Enterprise Storage Deployments
[4] Protocol Aware Recovery

TigerBeetle's Design Decisions

- strictly serializable
- highly durable and available [1]
  - network and node faults [2]
  - storage fault model[3] [4]
  - deterministic simulation testing and extensive fuzzing

"I want to emphasize, TigerBeetle has probably the strongest testing culture of any database I have worked with." Kyle Kingsbury @ SD25

[1] https://jepsen.io/analyses/tigerbeetle-0.16.11
[2] Viewstamped Replication
[3] A Study of SSD Reliability in Large-Scale Enterprise Storage Deployments
[4] Protocol Aware Recovery

TigerBeetle's Design Decisions

- strictly serializable
- highly durable and available [5]
  - network and node faults [6]
  - storage fault model [7]
  - deterministic simulation testing and fuzzing.
- no dependencies, static memory allocation, single core, asserts in production
- optimize for write-heavy and skewed workloads

[5] https://jepsen.io/analyses/tigerbeetle-0.16.11
[6] Viewstamped Replication
[7] Protocol Aware Recovery

tail latency = user experience

Tail latency and parallel calls

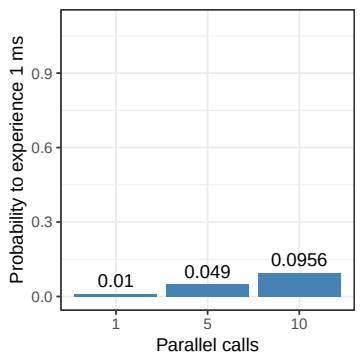
Assume P99 of a system is 1ms.

How high is the probability of hitting this latency with 1, 5, or 10 parallel requests?

Tail latency and parallel calls

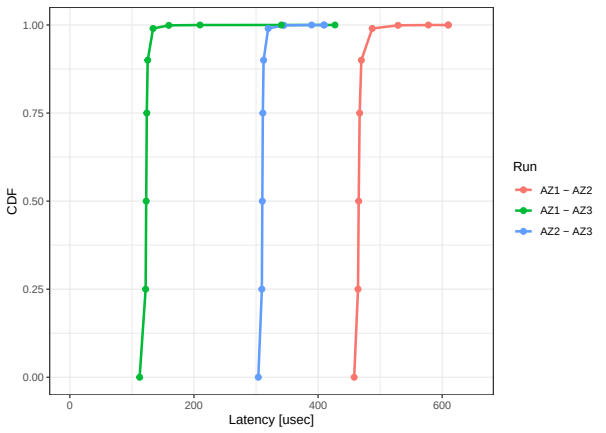
Assume P99 of a system is 1ms.

How high is the probability of hitting this latency with 1, 5, or 10 parallel requests?

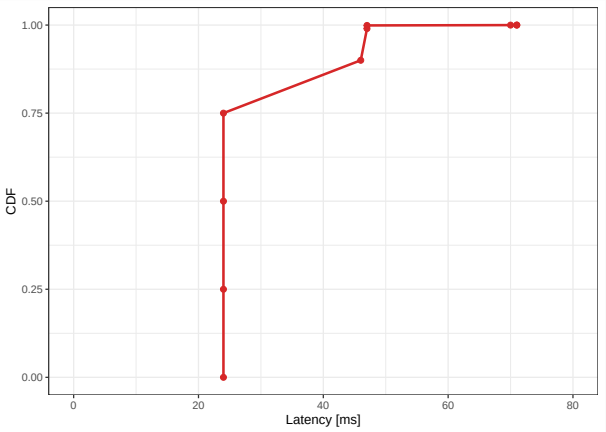


Act II: The Big Tail in Networks

The Network Jitters - Multi AZ



The Network Jitters - Cross Region



The Fallacies of Distributed Computing

1. The network is reliable
2. Latency is zero
3. Bandwidth is infinite
4. The network is secure
5. Topology doesn't change
6. There is one administrator
7. Transport cost is zero
8. The network is homogeneous

Viewstamped *Replication*

VR provides state machine replication.

Star Replication

Quorum Latency

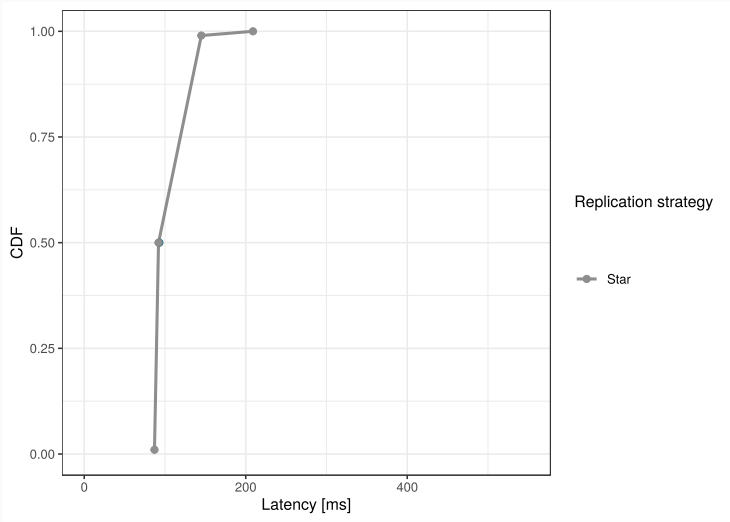
Latency	2^8	3	4	5
Mean	49	53	60	71
p50	48	48	60	71
p90	52	66	71	87
p95	58	71	71	94
p99	69	71	81	94
p99.9	71	75	93	105

Round-trip latency [ms] statistics for waiting on first N responses.

[8] H. Howard, Flexible Paxos: Quorum intersection revisited

Star Replication [3 regions, 6 x i4i.4xlarge]

Execute 10M transfers.

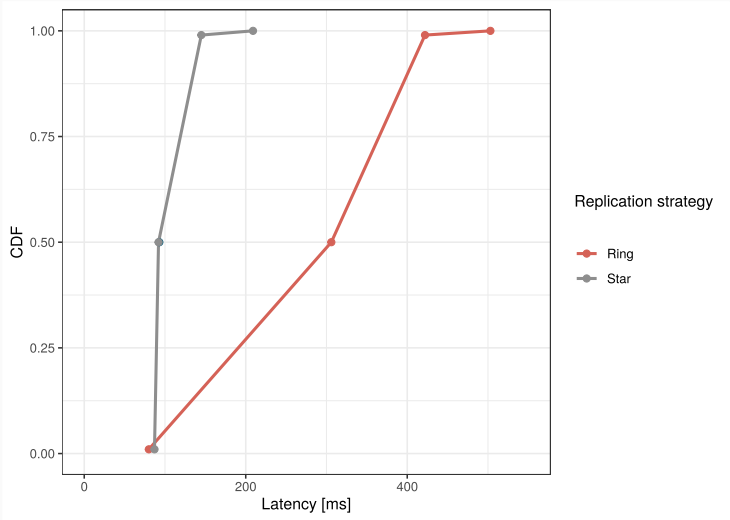


Star Replication

+ 1 RTT latency (for prepare)
+ masks failures (quorum)
- bandwidth consumption

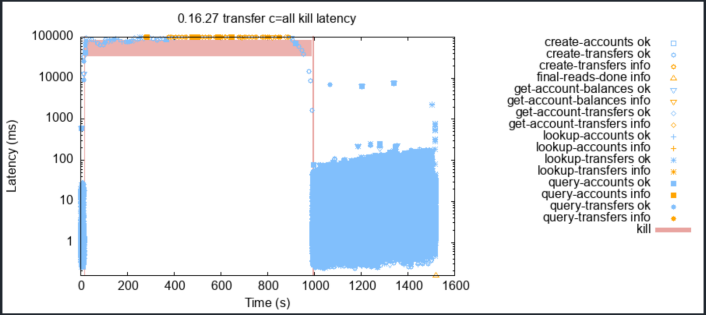
Ring Replication

Ring Replication



Ring Replication and Failures [Issue #2749]

In TigerBeetle 0.16.17 through 0.16.27, single-node failures are frequently associated with higher latencies across all clients--often from three to five orders of magnitude. For instance, consider this test. We killed one of three nodes, and saw latencies jump from 1-50 ms up to ~100 seconds. Latencies remained elevated until the node was restarted, almost a thousand seconds later.



High latencies may recover spontaneously after tens to hundreds of seconds, or persist for thousands of seconds.

This may involve the failure of a non-leader node. It also seems more likely in smaller clusters, rather than large ones.

Ring Replication

+ bandwidth optimized
- latency is not great
- masking faults is harder and requires retries[9]

[9] Try again: The tools and techniques behind resilient systems

Can we find a better trade-off?

	Star	Ring	?
Latency	++	–	
Bandwidth	–	++	
Failures	++	–	

Hybrid Replication

Goal: Fast common case latency, no message critical, bandwidth more balanced.

Good, right?

The Fallacies strike back!

1. The network is reliable
2. Latency is zero
3. Bandwidth is infinite
4. The network is secure
5. Topology doesn't change
6. There is one administrator
7. Transport cost is zero
8. The network is homogeneous

Why is it bad?

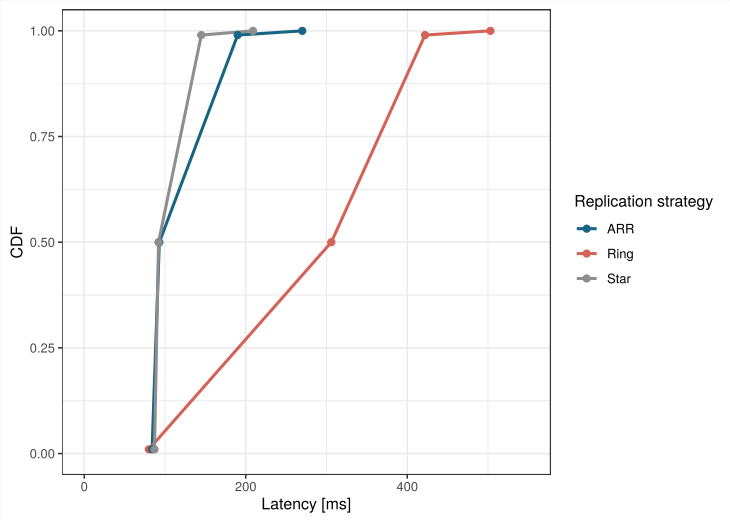
Gray Failure: The Achilles' Heel of Cloud-Scale Systems

Adaptive Replication Routing

Inspired laterally by congestion control[10]

[10] PCC: Re-architecting Congestion Control for Consistent High Performance

Adaptive Replication Routing



Taming The Tail But at What Cost?

	Star	Ring	ARR[11]
Latency	++	–	+
Bandwidth	–	++	+
Failures	++	–	+
Topology	++	–	++
Transport Cost	–	0	++

[11] Fuzzer master class: https://tigerbeetle.com/blog/2025-11-28-tale-of-four-fuzzers/

Remember the Fallacies when building distributed systems.

Act III: Tail Latency Inside One Beetle

P99 CONF

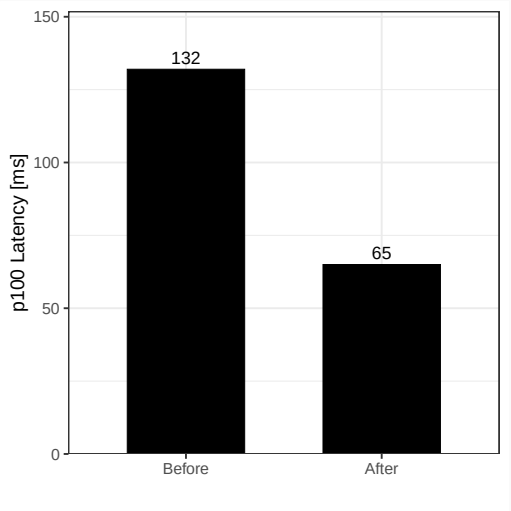
A ScyllaDB Community

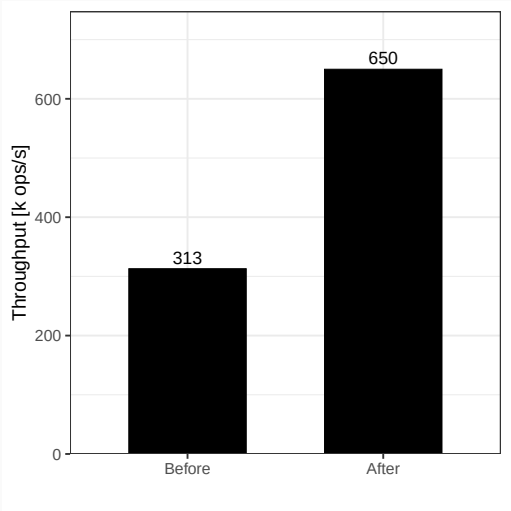
The Tale of Taming TigerBeetle's Tail Latency

Tobias Ziegler
Software Engineer

TigerBeetle

Results


150
100
50
0
p100 Latency [ms]
132
65
Before
After


600
400
200
0
Throughput [k ops/s]
313
650
Before
After

Recording

P99 CONF 2025 | The Tale of Taming TigerBeetle's Tail Latency by Tobias Ziegler