HW 10

KV – Store

Yue Wang, Hao Niu

Test Parameters:

- Duration: 60 seconds per test
- Concurrency: 10 workers
- Total test time: ~16 minutes (all configurations)

Performance Metrics Summary

Read Latency (Mean)

Configuration	1%/99%	10%/90%	50%/50%	90%/10%
W=5, R=1	0.78ms	0.81ms	0.51ms	0.74ms
W=1, R=5	0.69ms	0.79ms	0.61ms	0.74ms
W=3, R=3	0.71ms	0.79ms	0.54ms	0.63ms
Leaderless	0.70ms	0.75ms	0.55ms	0.66ms

Key Observations:

- All configurations show excellent read performance (< 1ms)
- Minimal variation between configurations
- Best performance: W=3, R=3 at 50%/50% (0.51ms)
- Worst performance: W=5, R=1 at 10%/90% (0.81ms)
- Difference is minimal (~0.3ms), all are very fast

Write Latency (Mean)

Configuration	1%/99%	10%/90%	50%/50%	90%/10%
W=5, R=1	303.78ms	303.50ms	304.28ms	305.24ms
W=1, R=5	303.40ms	304.96ms	305.82ms	306.58ms
W=3, R=3	304.89ms	304.90ms	304.55ms	305.13ms
Leaderless	303.35ms	303.43ms	304.53ms	306.04ms

Key Observations:

- All configurations show consistent write latency (~303-307ms)
- Variation is minimal (~4ms range)
- Best performance: Leaderless at 1%/99% (303.35ms)
- Worst performance: W=1, R=5 at 90%/10% (306.58ms)
- Write latency dominated by replication delays, not strategy

Read Latency Percentiles (P95)

Configuration	1%/99%	10%/90%	50%/50%	90%/10%
W=5, R=1	1.23ms	1.48ms	1.06ms	2.06ms
W=1, R=5	0.89ms	1.34ms	1.36ms	1.82ms
W=3, R=3	0.94ms	1.49ms	1.11ms	1.32ms
Leaderless	0.91ms	1.35ms	1.12ms	1.49ms

Long Tail Analysis:

- P95 values range from 0.89ms to 2.06ms
- Minimal long tail observed
- Highest P95: W=5, R=1 at 90%/10% (2.06ms)
- Lowest P95: W=1, R=5 at 1%/99% (0.89ms)

Write Latency Percentiles (P95)

Configuration	1%/99%	10%/90%	50%/50%	90%/10%
W=5, R=1	305.71ms	305.43ms	307.22ms	308.24ms

Configuration	1%/99%	10%/90%	50%/50%	90%/10%
W=1, R=5	304.34ms	307.77ms	309.06ms	310.12ms
W=3, R=3	306.62ms	307.27ms	307.26ms	307.82ms
Leaderless	304.96ms	305.36ms	307.05ms	309.16ms

Long Tail Analysis:

- P95 values range from 304.34ms to 310.12ms
- Very tight distribution (minimal tail)
- P95 is only ~2-5ms above mean
- No significant long tail for writes

Stale Reads

Observation: 0 stale reads detected across all configurations.

Analysis:

- All configurations: 0 stale reads
- This suggests replication completes before reads occur
- May require:
  - Higher concurrency (more simultaneous requests)
  - Longer test duration
  - More aggressive timing (reads immediately after writes)
  - Network delays or node failures

Test Generator Explanation

How It Works

Our load test generator uses a **"local-in-time" key clustering algorithm**:

1. **Key Organization:**
   - 1000 keys organized into 100 clusters (10 keys per cluster)
   - Each cluster represents related keys
2. **Active Cluster Tracking:**

  - Maintains list of "active" clusters (accessed in last 5 seconds)
  - Tracks last access time per cluster
  - Expires clusters after 5 seconds of inactivity
3. **Selection Algorithm:**
  - **80% probability**: Select from active clusters (recently accessed)
  - **20% probability**: Start new cluster
  - Within cluster: random key selection
4. **Temporal Locality:**
  - Keys accessed recently stay "active"
  - High probability (80%) of reusing active keys
  - Creates bursts of activity on same keys

How It Guarantees Local-In-Time Clustering

1. **80/20 Probability Split:**
  - 80% of requests target recently accessed keys (within 5 seconds)
  - 20% explore new keys
  - Ensures reads/writes to same key occur close together
2. **Active Cluster Window (5 seconds):**
  - Keys accessed in last 5 seconds are "active"
  - Multiple keys in same cluster accessed within this window
  - Creates temporal clustering
3. **Cluster Expiration:**
  - Prevents indefinite clustering
  - Allows exploration while maintaining locality
  - Realistic workload patterns
4. **Result:**
  - Reads and writes to same key typically occur within 5-second windows
  - Creates opportunities for stale reads
  - Triggers "return most recent value" logic
  - Demonstrates inconsistency windows

Example:

Time 0.0s: Write key_5 (cluster 0) → cluster 0 active

Time 0.1s: Read key_7 (cluster 0) → 80% chance, cluster 0 active

Time 0.2s: Write key_3 (cluster 0) → 80% chance, cluster 0 active

Time 0.3s: Read key_9 (cluster 0) → 80% chance, cluster 0 active

Time 5.1s: Cluster 0 expires

Time 5.2s: Write key_15 (cluster 1) → 20% chance, new cluster

Configuration Performance Analysis

Best Configuration by Workload Type

Read-Heavy (1% writes, 99% reads)

Winner: W=1, R=5 (0.69ms read mean, 0.89ms P95)

Analysis:

- Fastest read latency
- Writes are rare, so write latency less important
- Eventual consistency acceptable for read-heavy workloads

Alternative: Leaderless (0.70ms read mean, 0.91ms P95)

- Nearly identical performance
- No single point of failure
- Good choice for distributed systems

Moderate Read (10% writes, 90% reads)

Winner: Leaderless (0.75ms read mean, 1.35ms P95)

Analysis:

- Best read performance
- Good write performance (303.43ms)
- Balanced approach

Alternative: W=5, R=1 (0.81ms read mean, 1.48ms P95)

- Strong consistency
- Slightly slower reads

Balanced (50% writes, 50% reads)

Winner: W=3, R=3 (0.54ms read mean, 1.11ms P95)

Analysis:

- Best read performance at this ratio
- Good write performance (304.55ms)
- Quorum provides fault tolerance
- Balanced consistency and performance

Alternative: W=5, R=1 (0.51ms read mean, 1.06ms P95)

- Nearly identical performance
- Strong consistency

Write-Heavy (90% writes, 10% reads)

Winner: W=3, R=3 (0.63ms read mean, 1.32ms P95)

Analysis:

- Best read performance
- Good write performance (305.13ms)
- Quorum balances consistency and performance

Alternative: W=1, R=5 (0.74ms read mean, 1.82ms P95)

- Fast writes (but similar to others)
- Eventual consistency

Key Findings and Discussion

1. Read Performance is Excellent Across All Configurations

Finding: All configurations show sub-millisecond read latency (0.51-0.81ms mean).

Why:

- Reads are local operations (R=1) or well-optimized (R=5, R=3)
- No significant coordination overhead
- Network latency is minimal (localhost)

Implication: Read performance is not a differentiating factor between configurations.

2. Write Latency is Consistent Across Strategies

Finding: All configurations show similar write latency (~303-307ms).

Why:

- Write latency dominated by replication delays (200ms + 100ms per node)
- Strategy differences are minimal compared to delay overhead
- Concurrent replication requests reduce perceived latency

Implication: Write strategy choice has minimal impact on latency in this setup.

3. Minimal Long Tail Observed

Finding: P95 and P99 values are close to mean, indicating minimal long tail.

Why:

- Consistent replication delays
- No network variability (localhost)
- No node failures or congestion

Implication: In production with network variability, long tails may be more pronounced.

4. No Stale Reads Detected

Finding: 0 stale reads across all configurations.

Why:

- Replication completes before reads occur
- Test timing may not catch inconsistency windows
- Client-side version tracking may need adjustment

Implication: Stale reads may require:

- Higher concurrency
- Longer test duration
- More aggressive timing
- Network delays

5. Time Intervals Confirm Local-In-Time Clustering

Finding: Time interval graphs show clustering of reads/writes to same key.

Why:

- Test generator algorithm working as designed
- 80% probability of using active clusters
- 5-second active window creates temporal locality

Implication: Test generator successfully creates local-in-time patterns.

Recommendations

For Read-Heavy Applications (1%/99%, 10%/90%)

Best Choice: W=1, R=5 or Leaderless

- Fast read performance
- Eventual consistency acceptable
- No single point of failure (Leaderless)

For Balanced Applications (50%/50%)

Best Choice: W=3, R=3

- Excellent read performance
- Good write performance
- Fault tolerance (quorum)
- Balanced consistency

For Write-Heavy Applications (90%/10%)

Best Choice: W=3, R=3 or W=1, R=5

- Good performance for both operations
- W=1, R=5: Fast writes, eventual consistency
- W=3, R=3: Balanced approach with fault tolerance

General Recommendations

1. **Consistency Requirements:**
    - Strong consistency needed → W=5, R=1
    - Eventual consistency acceptable → W=1, R=5 or Leaderless

2. **Fault Tolerance:**
    - Need fault tolerance → W=3, R=3 (quorum)
    - No single point of failure → Leaderless
3. **Performance:**
    - All configurations show excellent performance
    - Differences are minimal
    - Choose based on consistency and fault tolerance needs