

FINAL PRESENTATION

SYSTEM BOTTLENECK ANALYSIS

Department of Computer Engineering

Presented by: Monil Manish Desai

Roll No: 25M0744

SYSTEM ARCHITECTURE

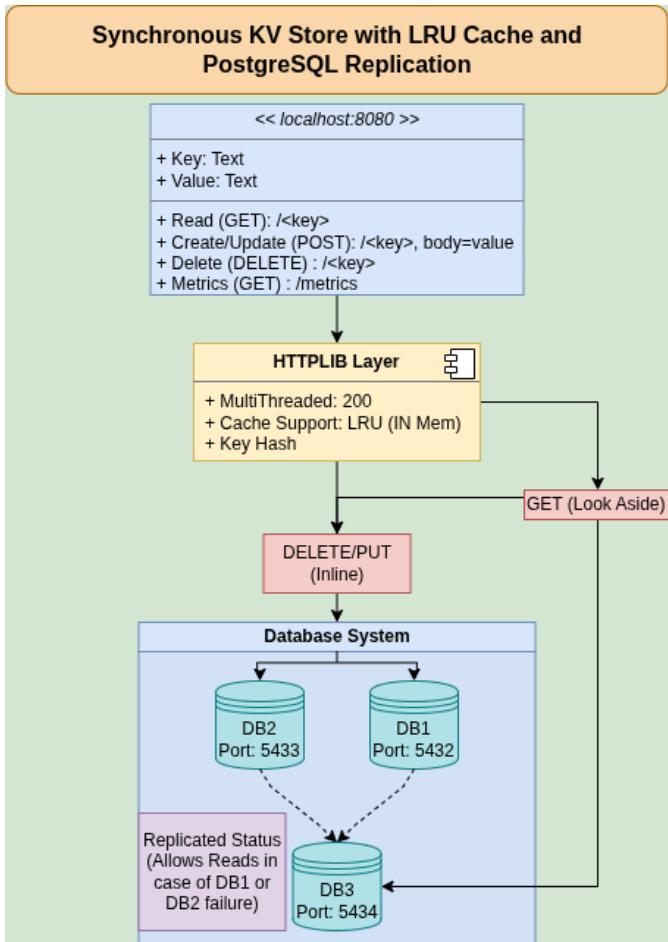


Fig 2.1: Multithreaded Server with LRU Cache & Sharded DBs

Core Components

- **HTTP Server:** Built using **cpp-httplib** (200 threads). Non-blocking I/O prevents stalls during DB writes.
- **LRU Cache:** In-memory storage for 10 hot keys. Absorbs repeated read traffic to reduce DB load.
- **PostgreSQL Backend:** Three isolated instances to prevent resource contention:
 - DB1 (Partition 1): Pinned to CPU 0-1
 - DB2 (Partition 2): Pinned to CPU 2-3
 - DB3 (Replica): Pinned to CPU 15

Mechanisms

- **Replication:** Async logical sync (`DB1/DB2 → DB3`) for redundancy.
- **Partitioning:** Client-side hashing routes writes to `DB1` or `DB2` (Sharding).

DEVELOPMENT & LOAD GENERATOR

Supported Workloads

- **get_popular1**

Read-heavy (10 hot keys)

→ Tests CPU/Cache

- **put_all**

Write-heavy (random keys)

→ Tests I/O Bandwidth/Disk

- **put_all_big**

Large Payloads (20KB)

→ Tests Memory/DRAM

- **put_key_1**

Single key writes

→ Tests Row-Level Locking

Load Generator Details

Custom **Closed-Loop Generator** (C++). Emulates real users via synchronous requests (Request → Wait → Next), creating natural backpressure.

```
./loader <threads> <time> <type>
```

Methodology

- **Experiments:** ~5 to 10 runs per bottleneck (T-1 to T-64).
- **Duration:** Fixed 5 minutes (300s) per run.
- **Metrics:**
 - Throughput (Reqs/sec)
 - Latency (Response ms)
 - Error Rate (% Failures)
 - **Utilization:** CPU, Disk I/O, Bandwidth

SETUP: WORKLOAD 1 (CPU BOUND)

Objective: Saturate Server CPU cores using a high-frequency read workload.

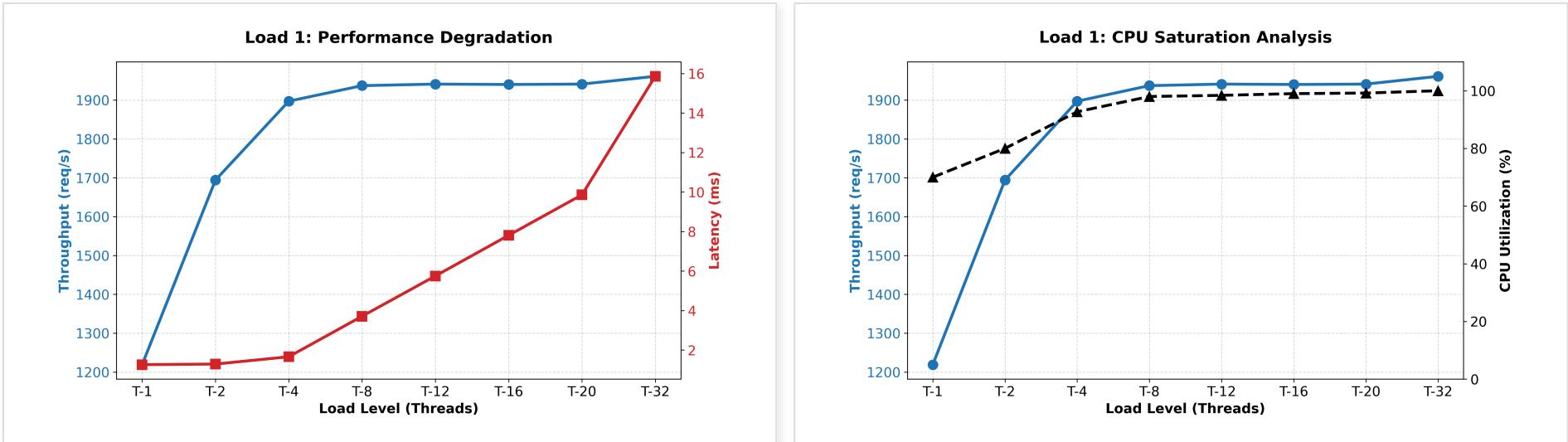
Hardware Topology

Component	Assigned Cores	Core Type
Server (Target)	4 - 5	P-Cores
Load Generator	6 - 14	Mix (P/E)
DB 1 & 2	0 - 3	P-Cores
DB 3	15	E-Cores

Configuration Parameters

- **Workload:** `get_popular1` (10 Hot Keys).
- **Server Config:** 200 Threads, Cache Size 10.
- **Constraints:** None (Unbounded CPU).
- **Goal:** Max Throughput via Cache Hits.

RESULTS: CPU SATURATION



Key Findings

- Throughput plateaus at ~1,960 req/s (T-8).
- **Correlation:** Plateau aligns perfectly with 100% CPU Utilization.
- **Latency:** Spikes (1.6ms → 15ms) as queue builds.
- *Note:* T-64 (Latency 32ms) excluded; confirms saturation.

SETUP: WORKLOAD 2 (BANDWIDTH)

Objective: Identify performance ceiling when Network/Bus is throttled.

Hardware Topology

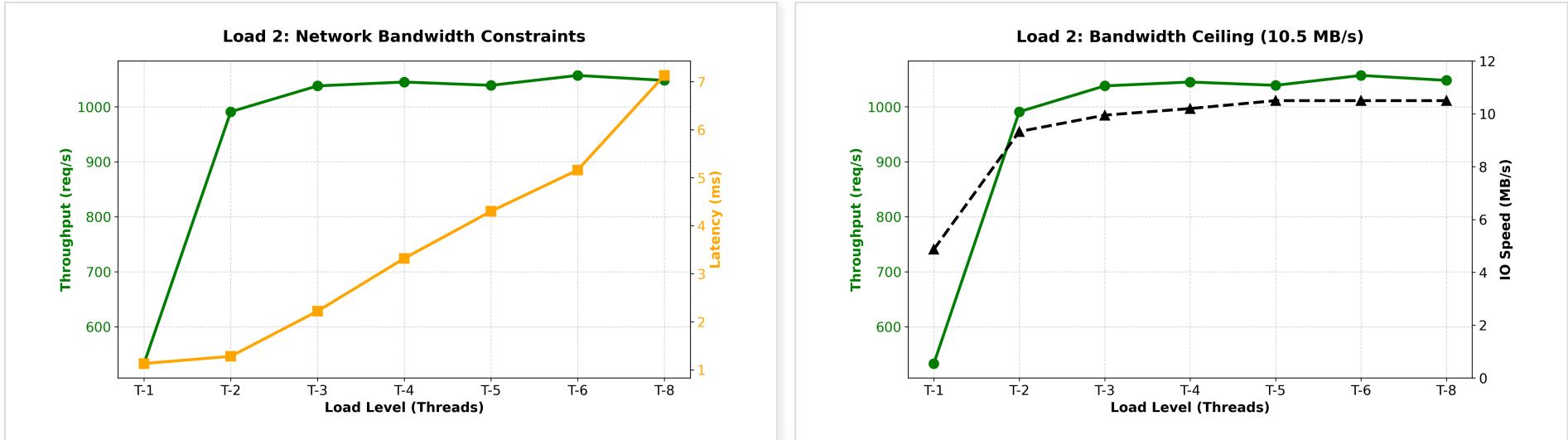
Component	Assigned Cores	Core Type
Server	4 - 11	P-Cores
Load Generator	12 - 14	E-Cores
DB 1 & 2	0 - 3	P-Cores
DB 3	15	E-Cores

Constraint (cgroups)

Throttling Block I/O to ~10.5 MB/s:

```
echo "7:0 rbps=10485760 wbps=10485760" | sudo tee /sys/fs/cgroup/system.slice/io.max
```

RESULTS: BANDWIDTH SATURATION



Key Findings

- **Hard Wall:** Throughput flatlines at ~1,050 req/s.
- IO Speed hits the 10.5 MB/s ceiling exactly at T-4.
- *Failure:* At T-10, buffers overflowed → "Fast-Fail" connection resets.

SETUP: WORKLOAD 3 (DISK I/O)

Objective: Test physical limits of storage (Max IOPS) with random writes.

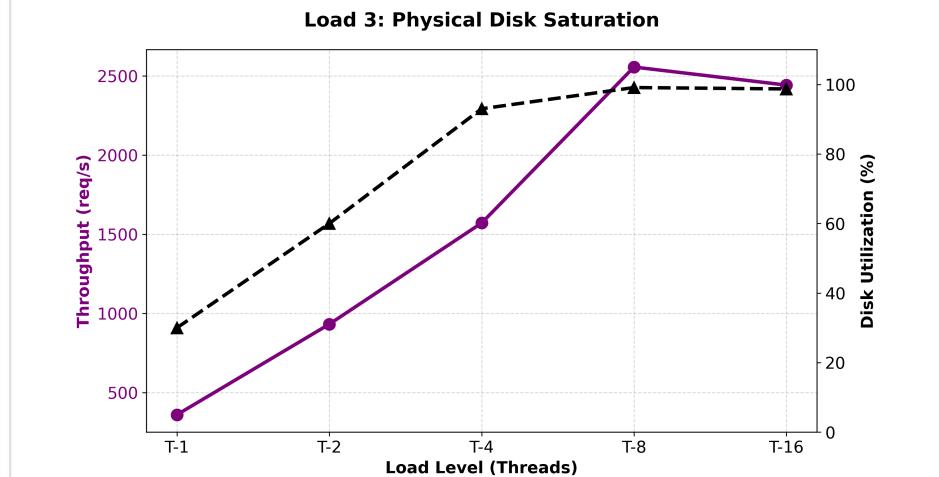
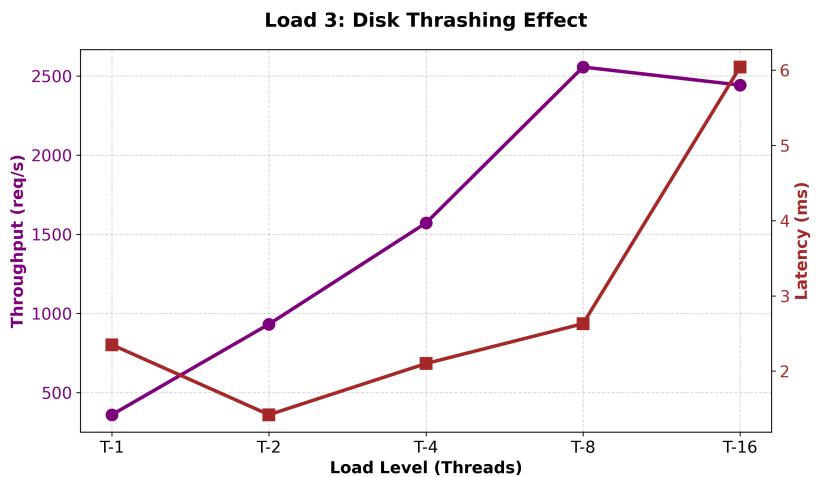
Hardware Topology

Component	Assigned Cores	Core Type
Server	4 - 11	P-Cores
Load Generator	12 - 14	E-Cores
DB 1 & 2	0 - 3	P-Cores
DB 3	15	E-Cores

Configuration

- **Workload:** `put_all` (Random Keys).
- **Constraint:** Removed (`rbps=max`).
- **DB Memory:** Capped 500MB (Forces Flush).

RESULTS: DISK I/O SATURATION



Key Findings

- Max Throughput: ~2,556 req/s at T-8 (99% Disk Util).
- Thrashing:** At T-16, throughput *drops* to 2,442 req/s.
- Queue management overhead outweighs parallel processing gains.

CONCLUSION

1. Read Scalability

Compute-bound. Scales linearly until CPU exhaustion.

2. Network Constraints

Creates a "Hard Ceiling". Fast-fail behavior masks failures.

3. Write Scalability

I/O bound. Susceptible to thrashing at high loads.

I hope the presentation latency was acceptable.
Thanks!!!