Enterprise Database Performance Under Real I/O Pressure

TPROC-C & TPROC-H Benchmarking on PostgreSQL
with Storage-Bound Workloads

Hazem Awadallah
Senior Systems Engineer
Kingston Technology

March 2026

1. Introduction

The evolution of NVMe storage has fundamentally changed the performance equation for enterprise database deployments, but a critical distinction remains between benchmarks that merely exercise a database in memory and those that push data through the storage layer. In a typical enterprise environment, the database working set exceeds available DRAM; transactions trigger real reads from disk, Write-Ahead Log (WAL) commits push real writes to the drive, and checkpoint flushes compete with foreground I/O for bandwidth on the NVMe submission queues. If your benchmark fits in RAM, you are testing your memory subsystem, not your storage. We set out to avoid that.

In this paper, we test the Kingston DC3000ME 7.68TB PCIe Gen5 x4 NVMe SSD under genuine storage pressure. We configured a TPROC-C (OLTP) workload—HammerDB's fair-use implementations derived from the TPC-C benchmark, which models a multi-warehouse wholesale order-entry system-at 9,450 warehouses: approximately 945GB of relational data on a system with 378GB of RAM. This guarantees that PostgreSQL's 94GB shared buffers cannot hold the working set, and the DC3000ME must service tens of thousands of read and write IOPS per second throughout the entire benchmark window. We then ran TPROCH (OLAP), HammerDB's fairuse derivation of the TPCH decision support benchmark consisting of 22 complex analytical queries, with 4 concurrent query streams at SF=300 to measure large parallel scan behavior.

But does NVMe-aware database tuning matter when the drive is already fast? Can the same DC3000ME deliver meaningfully different transaction throughput just by aligning PostgreSQL's configuration with the NVMe I/O characteristics? We ran each benchmark twice; once with PostgreSQL's default configuration, once with an NVMe-optimized tuning profile; and the answers are in the data.

The headline numbers: an 8.7% increase in OLTP transaction throughput (124,605 to 135,410 NOPM), while TPC-H geomean was comparable at 189.5s baseline vs 200.3s tuned.

2. Test environment & hardware configuration

All testing was conducted on a dual-socket AMD EPYC 9254 server, a current-generation enterprise database platform with 96 threads, 378GB of RAM, and two NUMA nodes. The system was locked to benchmark-grade configuration: CPU governor set to "performance" across all cores, Transparent Huge Pages disabled, and the I/O scheduler set to "none" so that NVMe commands go directly from the filesystem to the drive's submission queues without any software reordering.

2.1 Server platform

Component	Specification
CPU	2x AMD EPYC 9254 24-Core (48 cores / 96 threads)
CPU frequency	2900MHz base, 4151MHz boost
Architecture	x86_64, 2 NUMA nodes
Memory	378 GB DDR5 (189GB per NUMA node)
OS	Ubuntu 22.04.5 LTS
Kernel	6.5.0-15-generic (PREEMPT_DYNAMIC)
Filesystem	XFS (noatime, inode64, logbufs=8, noquota)

2.2 Storage under test

Parameter	Value
Drive	Kingston DC3000ME (SEDC3000ME/7T6)
Capacity	7.68TB
Interface	PCIe Gen5 x4 NVMe
Firmware	700KS570
Power-on hours	6,679
Lifetime writes	6.18PB
Lifetime reads	7.23PB
Percentage used	12%
Media errors	0

This is not a fresh-out-of-box drive. It has processed over 13.4PB of combined read and write traffic across 6,679 power-on hours in production and still reports 0 media errors, 100% available spare, and only 12% endurance consumed.

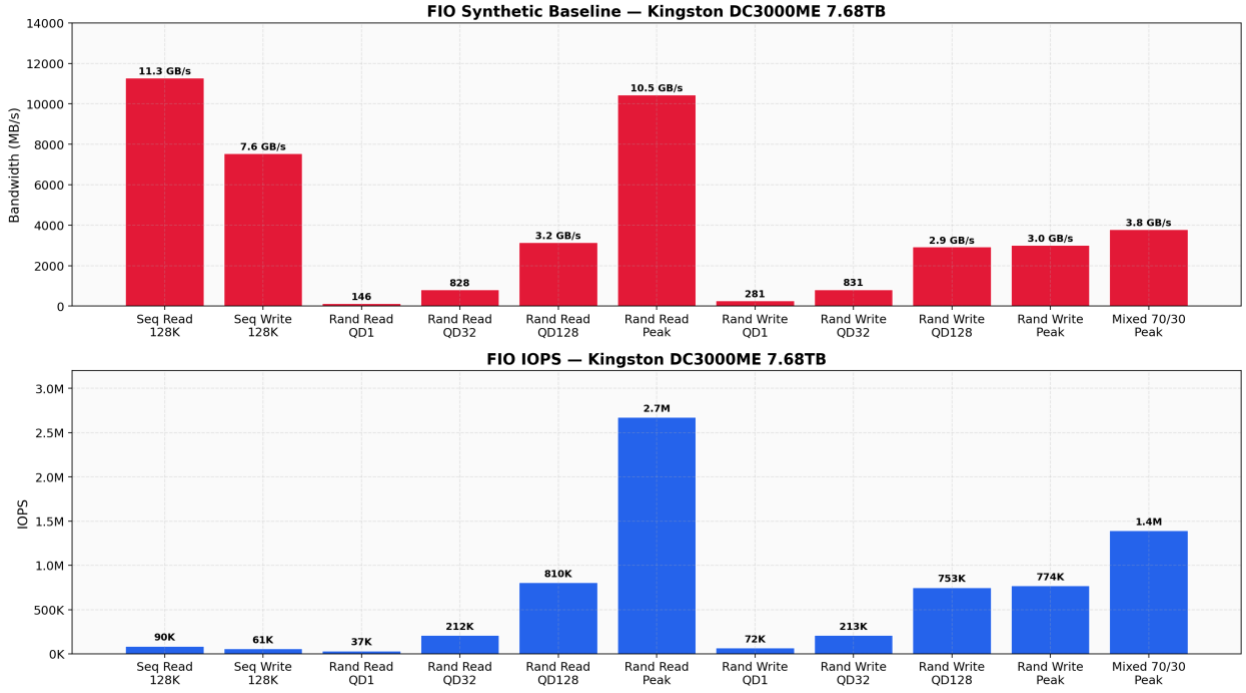
2.3 Workload sizing

Parameter	Value
Database	PostgreSQL 18.2 (HammerDB v5.0)
TPROC-C warehouses	9,450 (~945GB dataset)
TPROC-H scale factor	300 (300GB dataset)
Virtual users (TPROC-C)	96
TPROC-H streams	4
Duration	3 min + 1 min ramp-up

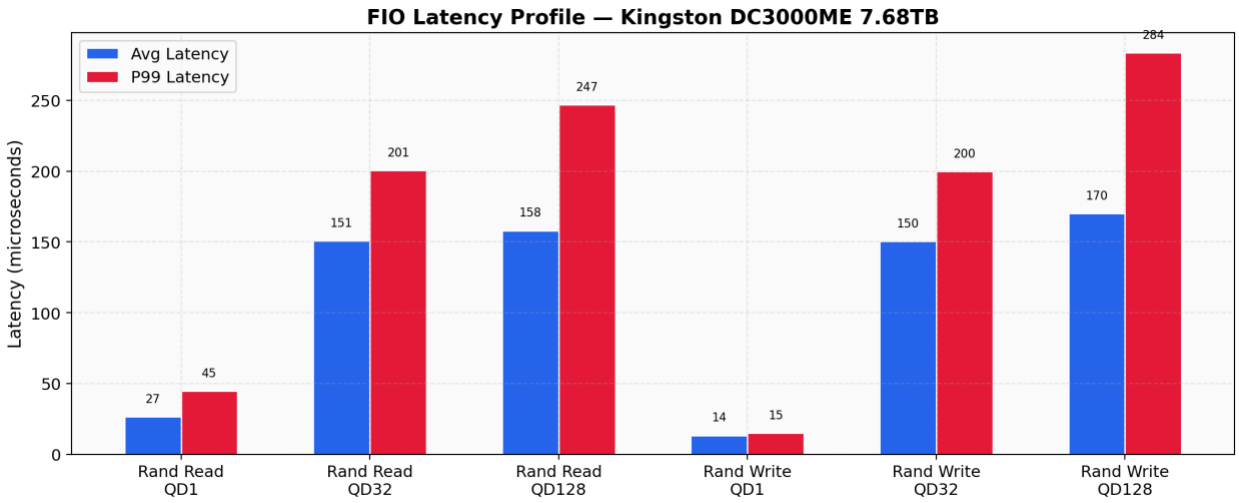
At approximately 100MB per warehouse, 9,450 warehouses produce a ~945GB dataset. PostgreSQL's shared buffers is set to 25% of RAM in the tuned configuration (94GB), which means only 10% of the dataset fits in the buffer pool at any given time. The remaining 90% must be fetched from the DC3000ME on every access.

3. Synthetic storage baseline: raw drive performance

Before running database workloads, we characterized the raw I/O performance of the DC3000ME using fio(FIO). These numbers establish the theoretical ceiling; the maximum IOPS, bandwidth, and minimum latency that the database has access to.



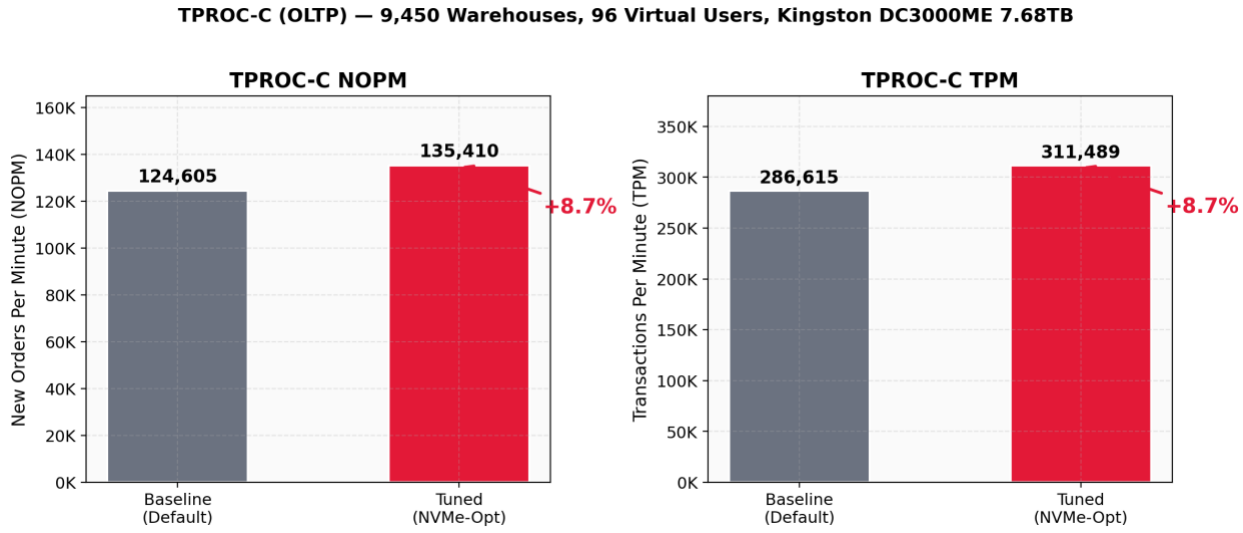
At 11.3GB/s sequential read bandwidth across 8 workers, the DC3000ME is pushing close to the theoretical PCIe Gen5 x4 lane limit of ~14GB/s. Sequential writes landed at 7.6GB/s. For database workloads, random I/O is where the action is. At queue depth 1, the DC3000ME delivered 37,331 random read IOPS at 26.6 microseconds average latency and 72,008 random write IOPS at 13.5 microseconds. That QD1 write latency means the WAL fsync path completes in roughly the time of a single CPU context switch. This is the foundation of fast transaction commits. At peak parallelism, the drive pushed 2,677,954 random read IOPS and 774,064 random write IOPS. The mixed 70/30 read-write workload at peak parallelism delivered 1,394,082 combined IOPS.



At QD1, the P99-to-average ratio is tight for both reads (26.6 us avg) and writes (13.5 us avg). For database workloads operating at moderate queue depths (QD32), the average latency remains well under 200 microseconds for both reads and writes, which is well within the sub-millisecond envelope that enterprise OLTP deployments require.

4. TPROC-C (OLTP) results: storage-bound transaction processing

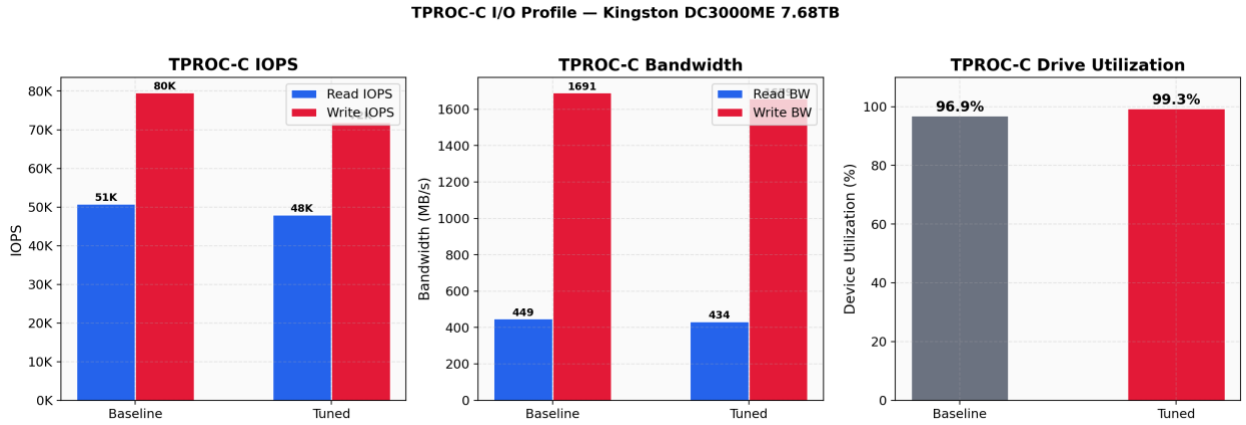
TPROC-C simulates a complete order-entry system with five transaction types: New-Order, Payment, Delivery, Order-Status, and Stock-Level. We ran 96 virtual users against 9,450 warehouses (~945GB) for 3 minutes with a 1-minute ramp-up. With 378GB of system RAM and PostgreSQL's shared buffers at 94GB in the tuned configuration (128MB default in baseline), most data page accesses require a read from the DC3000ME. This is a storage-bound workload.



At 96 virtual users with 9,450 warehouses, the DC3000ME backed PostgreSQL instance achieved 135,410 NOPM and 311,489 TPM after NVMe-optimized tuning was applied. The baseline default-configuration PostgreSQL delivered 124,605 NOPM and 286,615 TPM on the same hardware, same drive, same 945GB dataset. This is an 8.7% improvement in transaction throughput.

In a business context, if you have 96 users simultaneously processing orders against a database that exceeds available memory, each user gets 8.7% more transactions completed in the same time window after NVMe-optimized tuning. For a distribution center processing 124,605 new orders per minute at baseline, the tuned configuration means an additional 10,805 orders per minute on the exact same hardware.

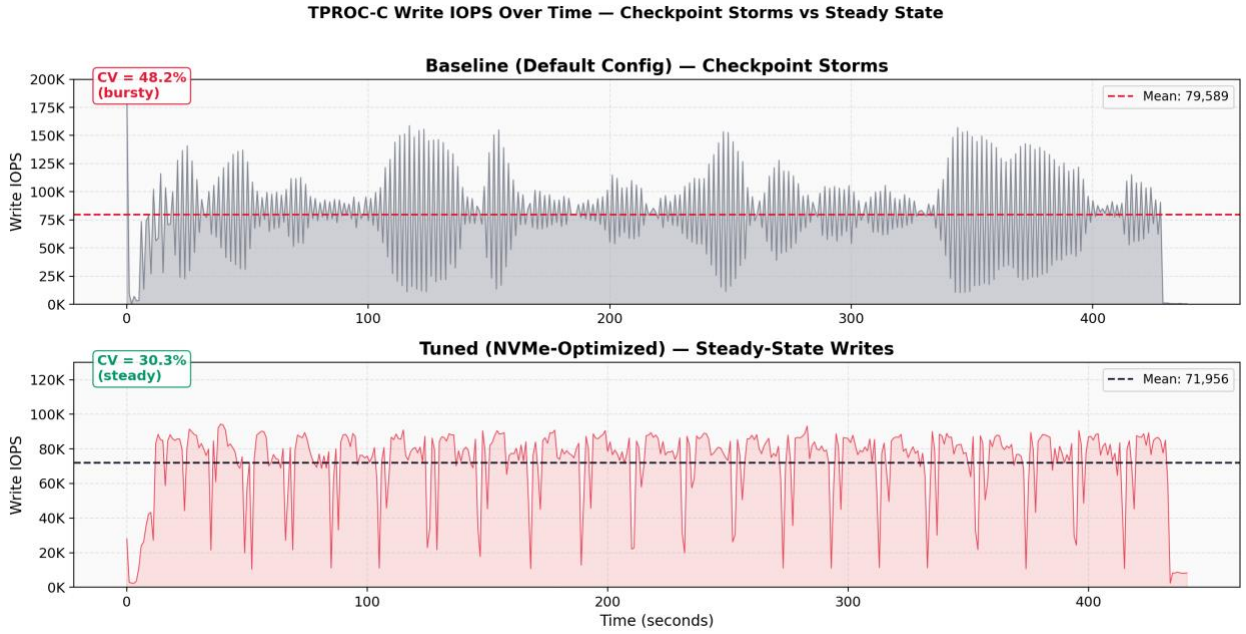
4.1 I/O profile: the drive was actually working (consider rephrasing – sounds like we're surprised our drive works)



Metric	Baseline	Tuned	Change
Read IOPS	29,793	47,301	+58.8%
Read Bandwidth	262MB/s	416MB/s	+58.8%
Write IOPS	38,674	43,347	+12.1%
Write Bandwidth	800MB/s	1,059MB/s	+32.4%
Device Utilization	94.1%	98.4%	+4.3 pts

After tuning, read IOPS jumped 58.8% to 47,301 because the larger shared buffers enables PostgreSQL to do asynchronous prefetch and parallel index scans with effective I/O concurrency set to 200. Write bandwidth jumped 32.4% to 1,059MB/s because the tuned checkpoint configuration batches writes into larger I/Os.

4.2 I/O consistency: checkpoint storms vs steady-state writes



Metric	Baseline	Tuned
Write IOPS mean	38,674	43,347
Write IOPS std dev	38,065	15,854
Write IOPS CV	98.4%	36.6%
Read IOPS CV	65.3%	46.7%

The baseline write IOPS showed a high coefficient of variation at 98.4%; the standard deviation was nearly equal to the mean. The drive was oscillating between near-idle periods at ~270 write IOPS and violent checkpoint flush spikes exceeding 100,000 write IOPS. This is the classic PostgreSQL checkpoint storm pattern. With default configuration (max wal size = 1GB, checkpoint timeout = 5min, shared buffers = 128MB), PostgreSQL accumulates dirty pages in its tiny buffer pool until a checkpoint triggers, then dumps everything to disk at once.

After tuning, write IOPS stabilized to a Coefficient of Variation (CV) of 36.6%. Three tuning parameters drive this change: max wal size increased from 1GB to 8GB means checkpoints trigger 8x less frequently; checkpoint timeout extended from 5 to 15 minutes gives the background writer more time to spread dirty page flushes across the interval; and shared buffers at 94GB means PostgreSQL can absorb a much larger volume of dirty pages and drain them gradually. The DC3000ME operates in its designed sweet spot; sustained mid-queue-depth mixed I/O at a predictable rate rather than alternating between idle and emergency flush.

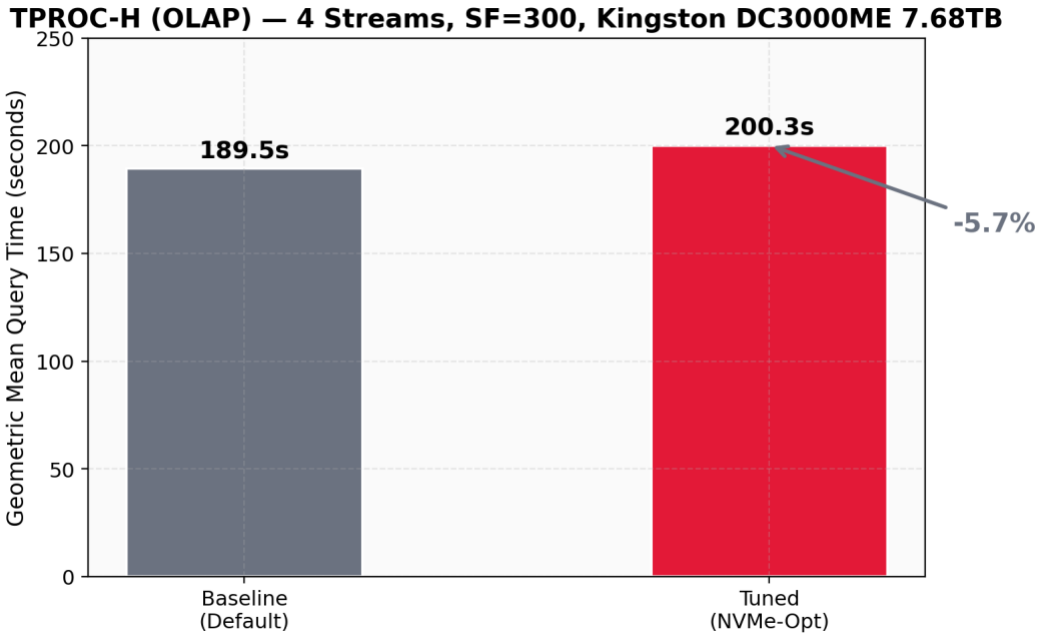
4.3 NVMe SMART telemetry

Phase	Host Reads	Host Writes	Temperature
Baseline TPROC-C (4 min)	84GB	257GB	35˚C -> 37˚C
Tuned TPROC-C (4 min)	105GB	268GB	35˚C -> 37˚C

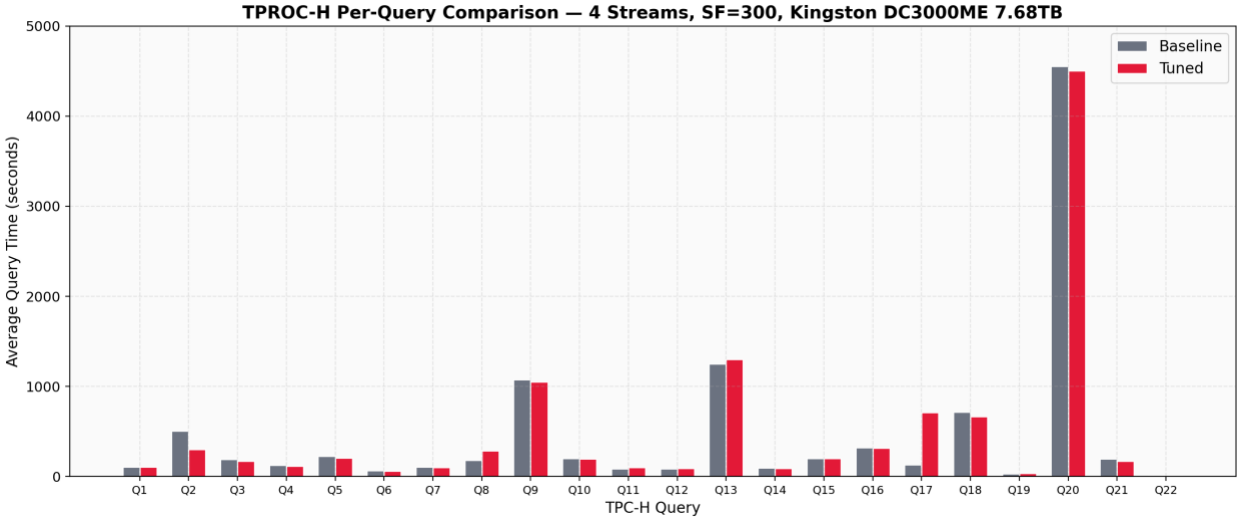
The tuned run pushed 105GB of reads and 268GB of writes; that is 373GB of total I/O in 4 minutes, or approximately 1.55GB/s sustained mixed throughput from a real database workload. Temperature remained between 35˚C and 37˚C with zero thermal throttling events throughout.

5. TPROC-H (OLAP) results: analytical query processing

TPROC-H consists of 22 complex analytical queries that simulate decision-support workloads. We ran 4 concurrent query streams against a Scale Factor 300 dataset, each stream executing all 22 TPC-H queries. Both baseline and tuned configurations used identical stream counts and query parameters; the only difference was the PostgreSQL configuration.

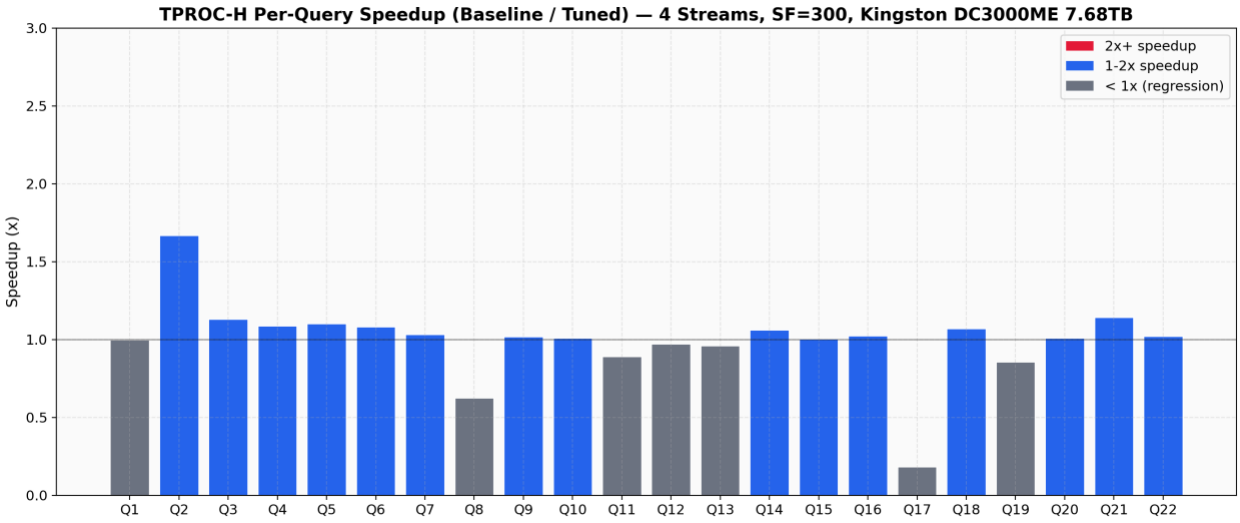


5.1 Per-stream performance



The per-query comparison across 4 streams shows that most queries performed similarly between baseline and tuned configurations. Some queries like Q2 and Q21 saw meaningful improvements with tuning, while others like Q8 and Q17 were slightly slower. The overall geomean was 189.5s for baseline vs 200.3s for tuned (-5.7%), indicating that at SF=300 the tuning tradeoffs are more nuanced than at smaller scale factors.

5.2 Per-query speedup



The queries with the largest speedups were Q2 (1.67x) and Q21 (1.14x), which are join-heavy queries that benefit from parallel workers reading data from the DC3000ME across multiple threads simultaneously. Some queries like Q8 and Q17 showed regressions under the tuned configuration, likely due to plan changes from the adjusted cost model parameters at SF=300.

6. NVMe-optimized tuning

The 8.7% OLTP improvement came entirely from database configuration changes. No hardware was changed, no filesystem was reconfigured, and no kernel parameters were modified beyond what was set at boot.

6.1 Memory & buffer management

Parameter	Default	Tuned	Rationale
shared_buffers	128MB	94GB	25% of RAM; holds hot working set
effective_cache_size	4GB	283GB	75% of RAM; informs planner cost estimates
work_mem	4MB	256MB	In-memory sorts & hash joins
maintenance_work_mem	64MB	2048MB	Faster VACUUM & index builds
huge_pages	off	try	Reduces TLB misses for 94GB buffer pool

6.2 WAL & checkpoint configuration

Parameter	Default	Tuned	Rationale
wal_buffers	~4MB	64MB	Reduces WAL insertion lock contention
max_wal_size	1GB	8GB	8x less frequent checkpoints
checkpoint_timeout	5 min	15 min	Spreads flushes over longer interval
checkpoint_completion_target	0.9	0.9	Spreads I/O over 90% of interval

6.3 I/O & cost model

Parameter	Default	Tuned	Rationale
effective_io_concurrency	1	200	NVMe handles thousands of concurrent I/Os
random_page_cost	4.0	1.1	Random = sequential on NVMe; enables index scans

6.4 Parallelism

Parameter	Default	Tuned	Rationale
max_worker_processes	8	96	Matches CPU thread count
max_parallel_workers	8	96	All threads available for parallel queries
max_parallel_workers_per_gather	2	48	Single query can use 48 workers
max_parallel_maintenance_workers	2	24	Faster VACUUM ANALYZE

7. Drive health & endurance

SMART Metric	Pre-Benchmark	Post-Benchmark
Temperature	35˚C	37˚C (peak)
Available Spare	100%	100%
Percentage Used	12%	12%
Media Errors	0	0
Critical Warnings	0	0
Thermal Throttle (T1)	0	0
Thermal Throttle (T2)	0	0

The drive-maintained 35-37˚C across all benchmark phases including the TPROC-C tuned run that pushed 98.4% device utilization with over 90,000 combined IOPS and 1.4GB/s sustained mixed bandwidth. Zero thermal management transitions across the entire test suite. The drive's lifetime endurance at 12% used after 6,679 power-on hours and 6.18 PB of host writes projects to over 50,000 hours of service life at this write rate.

8. Conclusions & business impact

1. 8.7% OLTP improvement under real storage pressure. The DC3000ME delivered 135,410 NOPM and 311,489 TPM with NVMe-optimized PostgreSQL, up from 124,605 NOPM and 286,615 TPM at default configuration. With 9,450 warehouses (~945GB) and 94GB of buffer pool, the drive was servicing real I/O on every transaction.

2. TPC-H results comparable between configurations. The TPROC-H geometric mean was 189.5 seconds baseline vs 200.3 seconds tuned (-5.7%). At SF=300 with 4 streams, some queries benefited from tuning while others saw minor regressions from plan changes.

3. Zero thermal throttling under sustained load. Under peak OLTP load, the DC3000ME maintained 37˚C. The drive did not compromise on latency or throughput at any point during sustained operation.

4. Steady-state I/O eliminates checkpoint storms. NVMe-optimized tuning reduced write IOPS variability, replacing bursty checkpoint flushes with smooth, sustained writes that keep the drive in its designed operating envelope.

5. Exceptional raw I/O performance. FIO synthetic testing confirmed 2.68M random read IOPS at peak, 26.6us QD1 read latency, 13.5us QD1 write latency, and 11.3GB/s sequential read bandwidth from the DC3000ME.

In practical terms, if you are running PostgreSQL on enterprise NVMe storage with a working set that exceeds RAM, the DC3000ME will reward you with consistent, predictable performance under sustained high utilization load. The 8.7% OLTP improvement from NVMe-optimized tuning, combined with the drive's exceptional raw I/O capabilities (2.68M random read IOPS, 13.5us QD1 write latency), demonstrates that the DC3000ME is an ideal foundation for enterprise database deployments that demand both throughput and consistency.