World Journal of Advanced Research and Reviews

eISSN: 2581-9615 CODEN (USA): WJARAI

Cross Ref DOI: 10.30574/wjarr

Journal homepage: https://wjarr.com/

(Review Article)

Disk profiling and cost optimization for latency-sensitive Cassandra DB (NoSQL DB)

Muthuselvam Chandramohan *

Leading Commercial Banking in USA.

World Journal of Advanced Research and Reviews, 2025, 26(02), 2973-2978

Publication history: Received on 03 April 2025; revised on 11 May 2025; accepted on 13 May 2025

Article DOI: https://doi.org/10.30574/wjarr.2025.26.2.1852

Abstract

This article explores the critical role of disk storage selection for optimizing Cassandra database performance in latency-sensitive environments. Beginning with an examination of IOPS measurement techniques, the discussion progresses through Cassandra's unique workload patterns, comprehensive disk profiling methodologies, rigorous benchmarking approaches, and cost-effective optimization strategies. By understanding the interplay between sequential writes, random reads, and mixed I/O patterns during compaction processes, database administrators can make informed decisions when selecting appropriate storage solutions. The article emphasizes the importance of evaluating sustained performance rather than relying on burst capabilities or vendor specifications, while highlighting how tiered storage approaches can balance performance requirements with budget constraints. Through systematic evaluation and workload-specific optimization, organizations can achieve optimal performance for NoSQL deployments while controlling infrastructure costs.

Keywords: NoSQL performance; IOPS measurement; Disk profiling; Tiered storage; Latency optimization

1. Introduction

In the era of real-time transactions and digital modernization, NoSQL databases are critical for delivering millisecond response times while managing massive data volumes. The choice of disk storage plays a pivotal role in ensuring an optimal, scalable, and cost-effective database infrastructure. Factors such as disk type, capacity, performance, and cost must be carefully evaluated to support high-throughput, low-latency applications.

The modern digital landscape demands database systems capable of handling extensive workloads with minimal latency. Cassandra DB, as described by Avinash Lakshman and Prashant Malik, was designed specifically to run on hundreds of nodes across multiple data centers with asynchronous masterless replication, providing highly available service with no single point of failure [1]. This architecture makes Cassandra particularly sensitive to storage performance characteristics, as it must manage a write-optimized storage engine with background compaction processes.

The implementation of efficient storage solutions is exemplified by streaming platform’s Cassandra deployment, which started with just 12 nodes in 2013 but grew substantially to support critical streaming services. By implementing careful disk profiling and optimization, streaming platform's infrastructure evolved to manage a Cassandra cluster with hundreds of nodes storing metadata in a 2.6TB dataset [2]. Their experience demonstrates how proper storage selection directly impacts the ability to maintain performance at scale, particularly for write-heavy workloads in distributed environments.

This article explores strategies for selecting the most efficient disk solutions for NoSQL databases, balancing performance requirements against cost considerations while ensuring seamless scalability for evolving digital

[*] Corresponding author: Muthuselvam Chandramohan.

ecosystems. By implementing structured approaches to storage evaluation and selection, organizations can achieve optimal performance for latency-sensitive workloads without unnecessary infrastructure expenditure.

2. IOPS Measurement: The Foundation of Disk Selection

Optimizing disk selection for Cassandra DB requires a structured approach, beginning with IOPS measurement. Disk IOPS significantly impact latency and concurrency, making it essential to measure actual performance using tools like BCC-tools (cachestat, biolatency, xfsslower, xfsdist), eBPF, and FIO rather than relying solely on vendor specifications. Selecting the right tool ensures accurate performance insights for informed decision-making.

Accurate IOPS measurement has become more sophisticated with the emergence of eBPF technology, which provides unprecedented visibility into storage performance. According to BrendanGregg's comprehensive work on BPF Performance Tools, traditional storage benchmarks often miss critical performance characteristics that affect database operations. The book details how biolatency, one of the BCC tools, can measure disk I/O latency distributions with microsecond precision across different operation types, revealing performance cliffs that vendor specifications frequently obscure [3]. This granular visibility is particularly valuable for Cassandra deployments, where occasional latency spikes can dramatically impact application performance even when average IOPS appear adequate.

When conducting storage evaluations for Cassandra, it's crucial to test with workloads that accurately reflect production patterns. Instaclustr's benchmarking research demonstrates this importance through their systematic approach to measuring node throughput for Apache Cassandra. Their testing revealed that the type of storage volume significantly impacted performance, with tests of AWS gp2 volumes showing performance degradation at around 3,000 IOPS per volume, while provisioned IOPS volumes maintained consistent performance up to their configured limits [4]. The benchmark testing used the Yahoo! Cloud Serving Benchmark (YCSB) with a mixed 50/50 read/write workload, demonstrating that a single i3.2xlarge instance with NVMe storage could achieve up to 38,680 operations per second, vastly outperforming equivalent gp2-based instances at just 13,561 operations per second.

The benchmarking methodology must incorporate realistic data sizes that account for Cassandra's compaction overhead. Instaclustr's findings highlight that storage performance during compaction can drop by 30-40% if the disk selection doesn't account for this workload, affecting overall cluster throughput [4]. Their testing framework incorporated varying data sizes to ensure realistic measurement of compaction impact, using datasets that grew to 300GB per node during extended testing periods.

When selecting measurement tools for IOPS evaluation, administrators should combine low-level kernel tracing provided by BCC-tools with application-specific benchmarking frameworks to capture both the raw storage capabilities and their impact on actual Cassandra performance. This multi-layered approach provides the most accurate foundation for storage decision-making, ensuring that selected disk solutions will deliver consistent performance under the complex I/O patterns that Cassandra generates in production environments.

Table 1 Cassandra Performance Comparison Across Storage Types [3, 4]

Storage Type	Operations Per Second (50/50 Read/Write)	Performance During Compaction (% Reduction)	Latency Measurement Precision
AWS NVMe (i3.2xlarge)	38,680	30-40%	Microsecond
AWS gp2	13,561	30-40%	
AWS Provisioned IOPS	Not specified	30-40%	

3. Understanding key cassandra workload patterns

Cassandra's distributed architecture creates unique disk access patterns that must be considered when selecting storage solutions. The database performs sequential writes for commit logs, random reads for queries, and both sequential and random I/O during compaction processes. These diverse patterns require storage solutions that can handle mixed workloads without performance degradation.

The distinctive I/O characteristics of Cassandra stem directly from its architecture as a distributed NoSQL database designed for high availability and scalability. As explained by Instaclustr, Cassandra uses a distributed architecture with a ring design that partitions data across multiple nodes, allowing it to distribute workloads evenly to prevent bottlenecks [5]. This architectural approach directly influences disk access patterns. Cassandra's write path first records operations sequentially in commit logs for durability, then stores data in memory structures called memtables, and finally flushes these to disk as immutable SSTables. This multi-stage process creates a complex I/O profile where sequential writes predominate in normal operations, but random I/O increases significantly during reads and maintenance operations.

The complexity of these workload patterns becomes particularly evident during scaling events and maintenance operations. According to Walmart Global Tech's analysis of Cassandra deployments, disk I/O patterns change dramatically during compaction processes [6]. Their production clusters exhibited normal read latencies of 3-5ms, but these increased to 15-20ms during heavy compaction activities. The engineering team observed that compaction processes could consume up to 25% of disk I/O capacity on normal workloads, but this increased to over 70% during peak traffic periods, creating potential for cascading performance issues if not properly managed.

This sensitivity to disk performance becomes especially critical when considering Cassandra's repair operations. Walmart's engineers documented that during repair processes, disk I/O can increase by 3-4 times compared to normal operations, with associated increases in disk usage of up to 40% [6]. Their analysis showed that repair operations on a 1TB node with standard SSDs took approximately 6 hours to complete, while the same operations on NVMe storage completed in just 2.5 hours, demonstrating the significant impact storage selection has on maintenance operations.

The interplay between these different I/O patterns creates unique challenges for storage selection. While Cassandra can handle up to 10,000 operations per second per node with proper storage configuration, inappropriate disk selection can reduce this by 50-80% during compaction or repair activities [6]. This performance variation underscores why understanding Cassandra's complex workload patterns is essential for selecting storage solutions that maintain consistent performance across all database operations, including both foreground client requests and background maintenance processes.

4. Disk profiling: categorizing storage options

Once IOPS is measured, disk profiling helps categorize available storage options, including NVMe, SSD, and general-purpose or provisioned IOPS disks. Some disks support cascading for higher IOPS, but incorrect configurations may lead to performance degradation. Evaluating sustained vs. max IOPS is crucial in determining the optimal storage configuration tailored to workload demands.

Comprehensive disk profiling requires systematic evaluation of storage technologies against Cassandra's workload patterns. Recent advancements in cloud infrastructure have expanded available options, particularly in the NVMe space. According to AWS, the latest EC2 R8g instances powered by Graviton4 processors feature up to 4TB of NVMe SSD storage with significantly enhanced I/O capabilities [7]. These instances deliver up to 1,000,000 IOPS with 60% better price/performance compared to previous generation instances. This performance tier represents a substantial improvement for Cassandra workloads, where the combination of compute power and storage performance directly impacts overall database responsiveness. The local NVMe storage on these instances provides an important advantage for Cassandra deployments, as the direct-attached nature eliminates network overhead that exists with network-attached storage options.

When profiling disks for Cassandra, understanding the relationship between storage performance and database operations is essential. DataStax documentation emphasizes that proper storage selection must account for Cassandra's unique compaction processes and read patterns [8]. Their guidelines note that SSTable compaction can consume significant I/O resources, with major compactions potentially using 50% or more of disk throughput. This activity directly impacts foreground operations, making sustained IOPS capabilities critical for consistent performance. The documentation further recommends configuring concurrent_compactors based on available I/O capacity, suggesting one compactor per physical disk or RAID volume to maximize throughput without overwhelming storage resources.

For provisioned IOPS solutions, the configuration of concurrent_reads and concurrent_writes settings must align with available disk performance. DataStax recommends setting concurrent_reads to 16 times the number of drives and concurrent_writes to 8 times the number of cores when using SSDs [8]. This guidance reflects the asymmetric performance characteristics of most storage systems, where read and write operations have different performance profiles that must be balanced against Cassandra's workload patterns.

When evaluating burst vs. sustained performance, disk profiling should incorporate extended load testing that mirrors production patterns. While many storage solutions offer impressive burst capabilities for short durations, Cassandra's continuous operation means sustained performance is typically more relevant. This is particularly important when considering general-purpose SSD options that may throttle performance after depleting I/O credits, potentially causing unexpected latency spikes during critical operations. Proper disk profiling ensures that selected storage solutions maintain consistent performance across all database operations, including both client requests and maintenance processes.

5. Benchmarking Methodology for nosql Performance

Implementing a comprehensive benchmarking methodology is essential for accurately assessing disk performance for Cassandra workloads. This should include testing read/write ratios that match production patterns, simulating both peak and sustained workloads, and measuring not just throughput but also latency distribution percentiles (p95, p99) which are critical for user experience in latency-sensitive applications.

Effective benchmarking for Cassandra deployments requires careful consideration of workload characteristics that accurately reflect production environments. Research by Tilmann Rabl et al. demonstrates that standard benchmarking approaches often fail to capture the complexity of NoSQL database performance under real-world conditions [9]. Their comparative study evaluated six NoSQL databases, including Cassandra, using the Yahoo! Cloud Serving Benchmark (YCSB) with varied workloads. The research revealed that Cassandra showed significantly different performance characteristics across workload types, achieving 7,000-9,000 operations per second for update-heavy workloads but only 2,000-3,000 operations per second for read-heavy scenarios on identical hardware. This performance variance highlights why benchmarking methodologies must incorporate workload patterns that match intended production usage rather than using generic test patterns.

The configuration of benchmarking parameters substantially impacts measurement accuracy. According to Cristina Evangelista's comprehensive analysis of Cassandra performance benchmarking, proper methodology must incorporate both short-term and long-term testing to capture performance characteristics accurately [10]. Their research demonstrated that Cassandra's throughput typically decreased by 15-20% after running for 24 hours compared to the first hour of operation due to compaction activities and memory pressure. Their findings showed that short benchmarks consistently overestimated performance, with 10-minute tests showing average latencies of 1.3ms compared to 2.8ms observed in extended 24-hour tests using identical hardware and workloads.

Table 2 Cassandra Performance Metrics Across Different Workload Types and Test Durations [9, 10]

Workload Type	**Operations Per Second**	**Average Latency (ms)**	**p95 Latency (ms)**	**p99 Latency (ms)**	**Performance After 24 Hours (% Decrease)**
Update-heavy	7,000-9,000	5-15	15-60	40-150	15-20%
Read-heavy	2,000-3,000	5-15	15-60	40-150	15-20%

Measurement of latency distributions rather than averages provides critical insights for latency-sensitive applications. Tilmann Rabl et al. found that while average latencies for Cassandra operations typically ranged from 5-15ms, the 95th percentile values were often 3-4 times higher, and 99th percentile measurements could be 8-10 times higher than averages [9]. Their testing demonstrated that storage I/O capabilities directly impacted these latency distributions, with higher-performance storage options reducing the gap between average and tail latencies. This relationship between storage performance and latency distributions makes comprehensive benchmarking essential for accurate disk selection.

The benchmarking methodology should also incorporate varying levels of concurrency to identify optimal configurations. Cristina Evangelista's research identified that Cassandra's performance scaling is not linear with thread count, showing optimal throughput with 32-64 client threads in their test environment, with performance declining when thread counts exceeded 128 [10]. Their work demonstrated that different storage configurations exhibited different concurrency characteristics, with NVMe-based solutions maintaining consistent latency at higher thread counts compared to traditional SSDs, which showed more significant latency degradation as concurrency increased.

6. Balancing Cost and Performance: Optimization Strategies

Cost optimization is essential to balance performance and budget. High-IOPS disks may be overkill and costly, while cheaper alternatives might fail to meet latency requirements. Benchmarking disk options against platform needs helps in selecting the most efficient, scalable, and cost-effective solution for Cassandra DB. Strategies include tiered storage approaches, where hot data resides on high-performance media while colder data moves to more economical options.

Finding the optimal balance between performance and cost requires systematic evaluation of storage options against actual workload requirements. Facebook's engineering team demonstrated this principle in their large-scale distributed caching infrastructure, where storage performance directly impacts overall system efficiency [11]. Their analysis showed that while higher-performance flash storage reduced average read latency from 10ms to 0.5ms compared to traditional storage, the per-gigabyte cost increased by approximately 15x. This cost differential required careful workload analysis to justify the investment. By implementing a selective approach where only the most performance-critical data was placed on flash storage, they reduced overall storage costs by 30% while maintaining performance SLAs. The team measured actual IOPS requirements during peak hours (12pm to 2pm) and found that many systems were significantly over-provisioned, with workloads requiring only 25-30% of the provisioned IOPS capacity during typical operation.

The proper benchmarking methodology is crucial for making informed cost-optimization decisions. Cooper et al.'s work on YCSB (Yahoo! Cloud Serving Benchmark) provides a framework for evaluating different storage options for Cassandra workloads [12]. Their research documented that read-heavy workloads (Workload B with 95% reads) showed only 15-20% performance improvement when upgrading from standard to high-performance storage, while write-heavy workloads (Workload A with 50% reads/writes) demonstrated up to 3x throughput improvement with the same upgrade. These workload-specific performance differences directly impact cost-optimization strategies, as organizations can target expensive storage investments only where they deliver substantial benefits.

The tiered storage approach leverages these performance characteristics to optimize costs. YCSB testing showed that when data access patterns follow a Zipfian distribution (as is common in many applications), just 20% of the data typically receives 80% of access operations [12]. This access pattern lends itself to tiered storage approaches where frequently accessed data is placed on high-performance media while less frequently accessed data uses more economical options. The research demonstrated that proper implementation of a two-tier storage approach could reduce storage costs by 40-60% compared to uniform high-performance storage, while maintaining latency within 10-15% of the all-premium configuration.

When implementing cost optimization strategies, organizations must consider not just hardware costs but also operational factors. Facebook's approach included evaluating the operational overhead of different storage configurations, finding that simpler storage systems with slightly higher hardware costs often resulted in lower total cost of ownership due to reduced maintenance requirements and higher reliability [11]. Their operational data showed that more complex storage configurations increased operational incidents by 2-3x, creating hidden costs that offset apparent hardware savings.

Table 3 Storage Optimization Strategies: Performance Gains and Cost Savings [11, 12]

Storage Strategy	Latency Reduction	Relative Cost Per GB	Performance Improvement
Traditional Storage	Baseline	1x	Baseline
High-Performance Flash	95% (10ms to 0.5ms)	15x	Varies by workload
Tiered Storage (Two-tier)	0-10% penalty vs. all-premium	Mixed	Within 10-15% of all-premium

7. Conclusion

Effective disk selection for Cassandra deployments requires a holistic approach that addresses both performance requirements and economic constraints. By implementing structured methodologies for IOPS measurement, understanding the unique access patterns of distributed NoSQL architectures, categorizing available storage options through comprehensive profiling, conducting rigorous benchmarking with appropriate workload simulation, and applying targeted optimization strategies, organizations can create database infrastructures that deliver consistent low-latency responses while maintaining budget discipline. The tiered storage approach offers particularly promising results, allowing systems to leverage high-performance media for frequently accessed data while utilizing economical options for colder data. As digital ecosystems continue to evolve and scale, this balanced approach to storage infrastructure becomes increasingly vital for maintaining competitive advantages in applications where millisecond response times directly impact user experience and business outcomes.

consistent low-latency responses while maintaining budget discipline. This methodical approach to disk selection provides a foundation for scalable, responsive Cassandra deployments that can adapt to growing workloads while avoiding unnecessary infrastructure costs.

References

[1] Avinash Lakshman and Prashant Malik, "Cassandra: a decentralized structured storage system," ACM SIGOPS Operating Systems Review, 2010. [Online]. Available: https://dl.acm.org/doi/10.1145/1773912.1773922

[2] Eran Stiller, "Netflix Built a Scalable Annotation Service Using Cassandra, Elasticsearch and Iceberg," Infoq, 2023. [Online]. Available: https://www.infoq.com/news/2023/02/netflix-annotations-cassandra/#:~:text=Netflix%20started%20with%20a%2012,data%20size%20of%202.6TB.

[3] BrendanGregg, "BPF Performance Tools," BrendanGregg, 2019. [Online]. Available: https://www.brendangregg.com/bpf-performance-tools-book.html

[4] Instaclustr, "Node throughput Benchmarking for Apache Cassandra® on AWS, Azure and GCP," Instaclustr, 2024. [Online]. Available: https://www.instaclustr.com/blog/node-throughput-benchmarking-for-apache-cassandra/

[5] Instaclustr, "Apache Cassandra: Features, architecture, and how to get started," Instaclustr, 2022. [Online]. Available: https://www.instaclustr.com/education/apache-cassandra/apache-cassandra-features-architecture-and-how-to-get-started/#:~:text=Cassandra%20uses%20a%20distributed%20architecture,workloads%20evenly%20to%20prevent%20bottlenecks.

[6] Sandeep Malik, "Avoid pitfalls in scaling your Cassandra cluster: lessons and remedies," Walmart Global Tech, 2017. [Online]. Available: https://medium.com/walmartglobaltech/avoid-pitfalls-in-scaling-your-cassandra-cluster-lessons-and-remedies-a71ca01f8c04

[7] Micah Walter, "New Amazon EC2 Graviton4-based instances with NVMe SSD storage," AWS, 2025. [Online]. Available: https://aws.amazon.com/blogs/aws/new-amazon-ec2-graviton4-based-instances-with-nvme-ssd-storage/

[8] DataStax, "Tuning the database schema," DataStax, 2025. [Online]. Available: https://docs.datastax.com/en/planning/oss/schema-tuning.html

[9] Tilmann Rabl et al., "Solving Big Data Challenges for Enterprise Application Performance Management," Proceedings of the VLDB Endowment, 2012. [Online]. Available: https://vldb.org/pvldb/vol5/p1724_tilmannrabl_vldb2012.pdf

[10] Cristina Evangelista, "Performance Modelling of NoSQL DBMS," Proceedings of the 2020 OMI Seminars Research Trends in Data Centre Operations, Selected Topics in Data Centre Operations, and Research Trends in the Internet of Things (PROMIS 2020), 2020. [Online]. Available: https://d-nb.info/1239737092/34

[11] Rajesh Nishtala et al., "Scaling Memcache at Facebook," 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI ’13), 2013. [Online]. Available: https://www.usenix.org/system/files/conference/nsdi13/nsdi13-final170_update.pdf

[12] Brian F. Cooper et al., "Benchmarking Cloud Serving Systems with YCSB," courses.cs, 2010. [Online]. Available: https://courses.cs.duke.edu/fall13/compsci590.4/838-CloudPapers/ycsb.pdf