

# Scaling object storage performance while reducing power and complexity

Object storage plays a central role across today’s data center workloads. It underpins modern data lakes, backup/restore, content delivery, and AI/ML dataset management.<sup>1</sup>

To quantify the impact of storage media type on S3 object-store performance, Micron Data Center Workload Engineering (DCWE) analyzed MinIO object storage performance using the Warp S3 benchmark with 4MB objects.<sup>2</sup>

The tests compared two storage building blocks with similar raw capacity: a single Micron® 6600 ION 245TB NVMe™ SSD (1x SSD) and an array of 16x 16TB data center HDDs (16x HDD) presented as a RAID-0/JBOD array.<sup>3</sup> These configurations are not intended to represent complete production deployments; instead, they serve as comparable building blocks to help readers understand how SSD- and HDD-based designs behave under the same object-store workload.

Across the compared workload mixes, the single 245TB SSD delivered significantly higher throughput, higher throughput per TB, and faster time to first byte.<sup>4</sup>



Micron 6600 ION SSD (E3.S, E1.S, U.2)

## Throughput and power efficiency, not overhead

By turning capacity and power into usable system throughput, Micron 6600 ION SSD-based object storage demonstrated how object-storage-backed AI workflows can scale under real-world constraints. The results below summarize measured performance and efficiency by workload with all maximal values.

Up to

**56x**

Aggregate throughput

A single 245TB Micron 6600 ION SSD showed 45x–56x the object store throughput of the HDD array at matched capacity.

**96x**

Lower time to first byte

A faster initial response (first byte) means downstream processing can start earlier, reducing idle time.

**435x**

Throughput per watt

Higher throughput per watt lets an object store move more data without increasing power.

**58x**

Throughput per TB

More usable performance from a comparable capacity footprint turns every terabyte into more productive throughput.

[micron.com/6600-ION](https://micron.com/6600-ION)

1. See “Object storage built to retrieve any amount of data from anywhere” ([aws.amazon.com](https://aws.amazon.com)), “Cloud Storage” ([docs.cloud.google.com](https://docs.cloud.google.com)), “Use various origins with CloudFront distributions” ([docs.aws.amazon.com](https://docs.aws.amazon.com)), and “The Use of Object Storage for Modern Data-Intensive Workloads Requiring High Performance” ([nutanix.com](https://nutanix.com)) for more background on object store uses.  
 2. See the MinIO homepage ([min.io](https://min.io)) to learn more about their products. The MinIO Warp benchmark measures real-world object storage performance using S3-compatible GET ([min.io](https://min.io)) and PUT ([min.io](https://min.io)) operations with 4MiB objects, representing typical object storage workloads.  
 3. Rated capacity. Unformatted. 1GB = 1 billion bytes. Formatted capacity is less.  
 4. See “Best Practices Design Patterns: Optimizing Amazon S3 Performance” ([docs.aws.amazon.com](https://docs.aws.amazon.com)) for more background on the importance of these storage features.

Modern AI infrastructure decisions are rarely made at a small scale, but data center architects often evaluate smaller building blocks as proof-of-concept (PoC) examples when designing larger systems. In that spirit, this document uses a 1x 245TB Micron 6600 ION SSD and 16x 16TB HDD configuration as repeatable reference building blocks to highlight the trade-offs architects must consider when choosing SSD- or HDD-based object storage for their own requirements.

This study evaluates two similar-capacity building blocks to show how SSD and HDD architectures differ across a range of object storage workloads. The goal is **not** to declare a universal “winner,” but to provide insight and architectural context so readers can decide which storage type best fits their workload mix, responsiveness targets, and power/space constraints.

Parameter	SSD configuration	HDD configuration
Drive type	Micron 6600 ION SSD 245TB	Capacity-focused, data center HDD
Capacity per drive	245TB	16TB
Interface	NVMe (PCIe® Gen5)	Seagate EXOS 18
Tested configurations	1x 245.76TB SSD (245TB) (1x SSD)	16x 16TB HDD (256TB) (16x HDD)

Table 1: Configurations overview

## Aggregate throughput analysis

Aggregate throughput measures the combined, concurrent I/O generated by real object storage workloads supporting AI pipelines, analytics, and large-scale data services. Starting here ensures the comparison reflects system-level behavior rather than best-case microbenchmarks.

Just as importantly, aggregate throughput provides a baseline for deeper architectural comparisons and their effects, since the two configurations deliver similar raw capacity.

The results shown in Figure 1 start to answer a foundational question: How efficiently does each architecture convert its capacity into usable bandwidth?

**Observed throughput difference in this building-block comparison:** Across the tested GET/PUT mixes, the 1x SSD delivered 45x–56x times as much throughput as the 16x HDD building block, quantifying how differently each architecture converted similar raw capacity into usable bandwidth under the same conditions.<sup>5</sup>

**Read-heavy mixes change the shape, not the takeaway:** At 90/10 GET/PUT (the most read-heavy workload tested), the 16x HDD improved to 187 MB/s versus 8,473 MB/s for the 1x SSD. This illustrated that, while read bias can lift HDD throughput, the SSD and HDD building blocks still operated in very different throughput envelopes under the same workloads.

With that context set, the sections that follow reveal how architectural choices increasingly matter as workloads scale and concurrency rises.

In subsequent sections, references to “GET/PUT” below each workload are dropped for brevity. All subsequent charts and analyses align with the GET/PUT mixes seen in Figure 1.

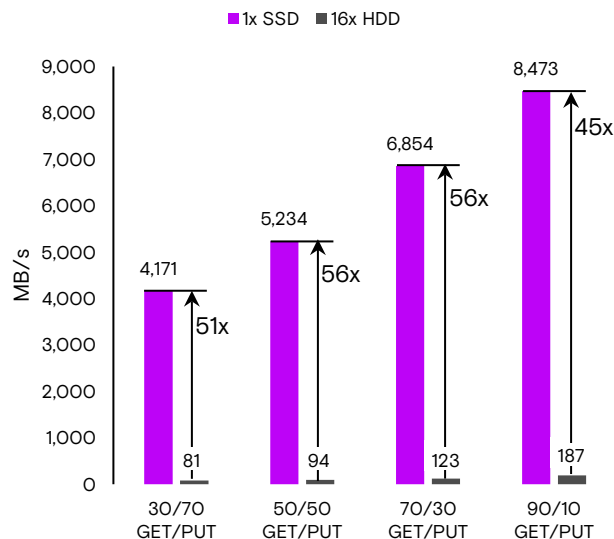


Figure 1: Aggregate throughput

5. All SSD performance advantages calculated as (1x SSD MB/s) / (16x HDD MB/s).

## Breaking down GET/PUT results

Next, we split aggregate throughput into GET and PUT to pinpoint what can shape performance across read/write mixes. Aggregate throughput shows how much work the system can deliver overall, but it does not explain why that performance emerges. This breakdown revealed how each storage architecture responded as workloads shifted from write-heavy ingestion to read-dominant access patterns typical of analytics and AI pipelines.

Figures 2 and 3 show these outcomes and highlight the respective architectural behaviors.

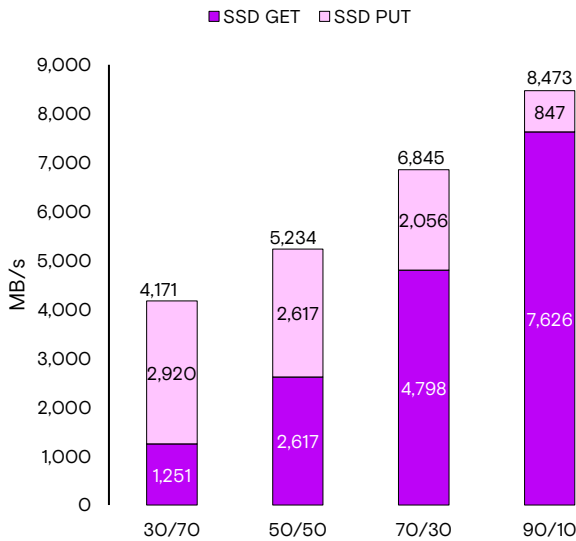


Figure 2: 1x SSD throughput mix

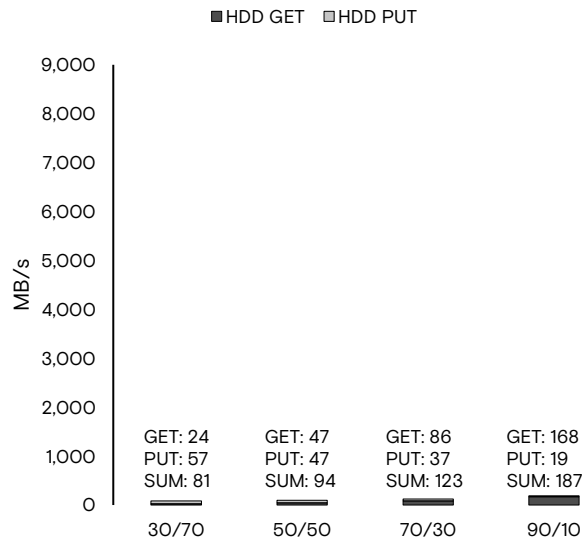


Figure 3: 16x HDD throughput mix

**GET and PUT both matter:** The 1x SSD lead was not just aggregate: at 30/70, its GET/PUT was 1,251 / 2,920 MB/s versus 16x HDD

24 / 57. At the opposite end (90/10), those values were 1x SSD: 7,626 / 847 versus 16x HDD: 168 / 19. A similar trend was observed for the remaining workloads as well.

**16x HDD best-case throughput was very low:** The 16x HDD was extremely limited across all mixes. For example, at 90/10, 1x SSD PUT = 847 MB/s, while the entire 16x HDD total was 187 MB/s (GET 168, PUT 19). The 1x SSD write throughput stayed meaningful under the workload read bias.

Breaking apart GET and PUT showed how each configuration responded as the mix shifted from write-heavy ingestion to read-dominant access. The 1x SSD sustained substantial throughput in both operations across all mixes, while the 16x HDD showed a much smaller GET/PUT envelope and a write path that collapsed under read bias.

Throughput shows how much work the system can move; latency shows how quickly a client sees progress. Therefore, next we moved to response-time metrics (including first-byte latency), because latency-sensitive applications can feel unresponsive even when aggregate GET and PUT rates are high.<sup>6</sup>

## Average request latency

Throughput defines how much work a storage system can perform, but latency determines how efficiently applications experience that work.

After establishing aggregate and workload-specific throughput, the next question is how quickly the system responds to individual requests under load. Average request latency provides that view, revealing how architectural choices affect responsiveness as workloads scale and concurrency increases.

6. See "Best practices design patterns: optimizing Amazon S3 performance" ([docs.amazon.com](https://docs.amazon.com)).

Figure 4 shows that the average GET latency results highlighted a fundamental architectural difference between the two configurations. The 1x SSD maintained consistently fast response times across all tested workload mixes, indicating predictable behavior as access patterns shifted. In contrast, the 16x HDD exhibited persistently high latency that changed little with workload composition, reflecting a fixed mechanical constraint rather than dynamic responsiveness.

A similar pattern was observed in average PUT latency, as shown in Figure 5. PUT latency analysis helps highlight how each architecture handles sustained data ingestion under mixed workloads. Across all read/write ratios, the 1x SSD consistently delivered lower average PUT latency, showing that write operations completed predictably as read demand increased. The HDD array maintained substantially higher write latency across workloads, increasing the likelihood of ingest backpressure as concurrency rises.

Average latency describes the typical responsiveness of each configuration, but it does not capture how often requests become stragglers. In fact, average latency can easily mask stragglers. In object storage, applications can fan

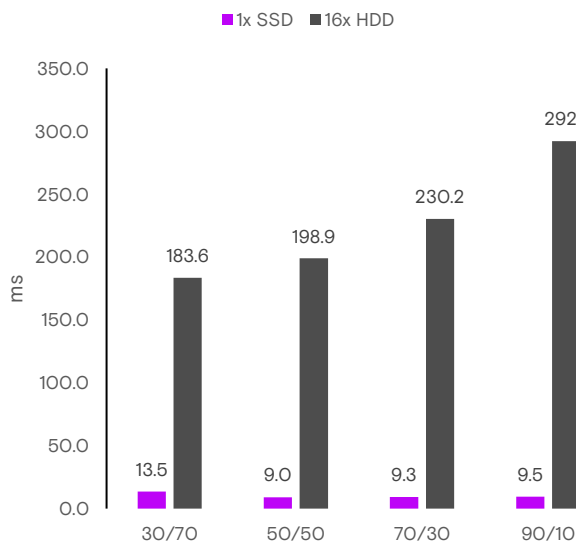


Figure 4: Average GET latency by workload

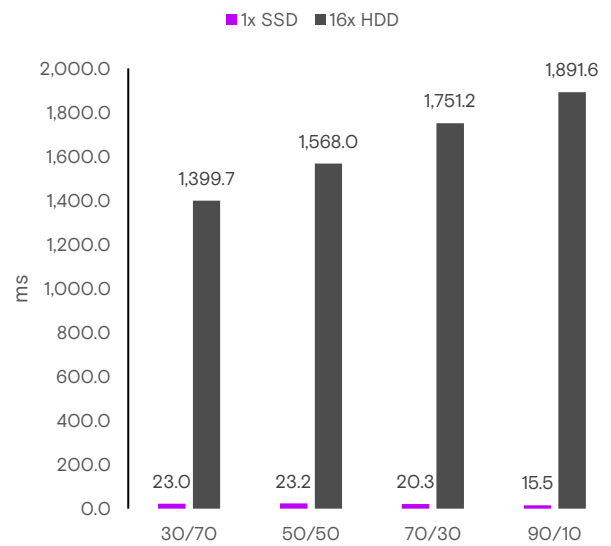


Figure 5: Average PUT latency by workload

out across many GET and PUT operations, making end-to-end completion time more sensitive to the slowest requests rather than the average.

## Tail request latency

Figures 6–9 show the P50 (50th percentile, or average latency), P90 (90th percentile), and P99 (99th percentile) GET latencies. The data shown in these figures reveals three architecture-focused insights:

- **SSD latency distributions stayed tight as workloads changed:** Across all workload mixes, 1x SSD showed modest separation between P50, P90, and P99 latency for both GETs and PUTs. This indicates that as concurrency increases, 1x SSD continued to service requests predictably, without producing stragglers.
- **HDD tail latency expanded sharply (even when averages appear stable):** While HDD average latency remained relatively flat across workloads, its P99 latency grew dramatically, particularly for PUT operations. This widening gap shows that averages mask a growing population of slow requests that increasingly govern application completion time.
- **Read-heavy workloads can magnify tail risk:** As workloads shifted toward read-dominant mixes (70/30 and 90/10), SSD tail latency tightened further, while HDD P99 latency remained high.

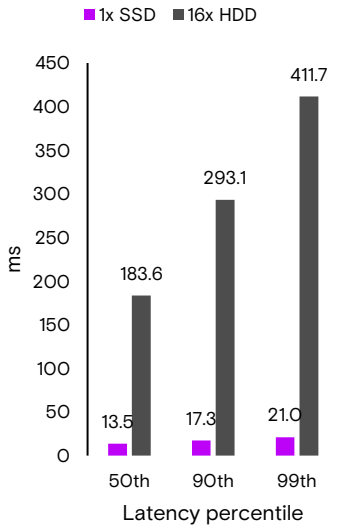


Figure 6: 30/70 GET/PUT tail latencies

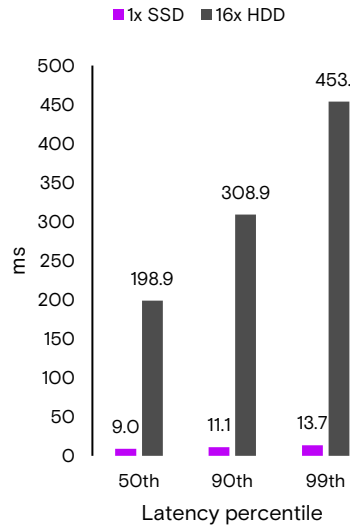


Figure 7: 50/50 GET/PUT tail latencies

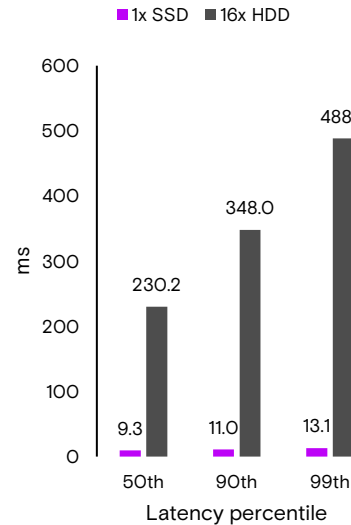


Figure 8: 70/30 GET/PUT tail latencies

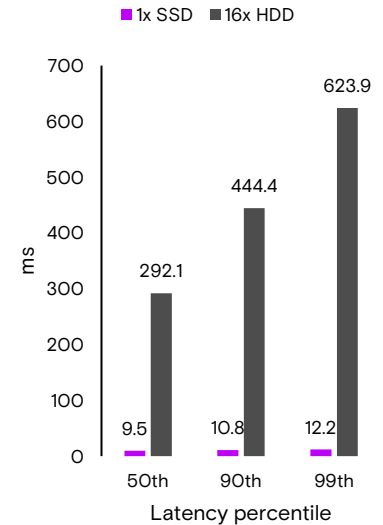


Figure 9: 90/10 GET/PUT tail latencies

## Time to first byte (TTFB)

In MinIO object storage object GET, we define TTFB as the time from issuing the request until the first byte of object data (payload) is available at the client (i.e., when the object begins streaming).<sup>7</sup> Lower TTFB is better because it helps reduce wait time before any progress is visible to the client.

Figure 10 shows that the 1x SSD TTFB advantage was significant at all tested workload mixes.

**First-byte delivery:** The 1x SSD TTFB ranged from 3ms to 5ms across all workloads, while the 16x HDD TTFB ranged from 180ms to 289ms. This gave the 1x SSD configuration a TTFB advantage of 36x to 96x.<sup>8</sup>

**Gaps widen:** The 16x HDD TTFB worsened as GET share rose (180→196→227→289 ms), while the 1x SSD TTFB remained flat (5→3→3→3 ms), widening the responsiveness advantage as GET share increases.

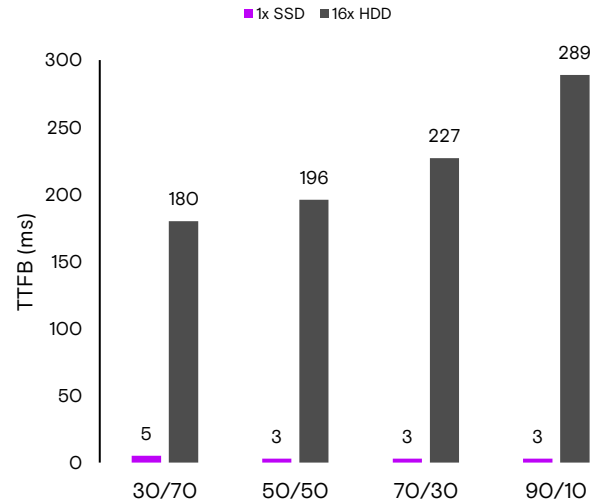


Figure 10: Time to first byte

## Throughput per watt

Storage value is not just determined by performance and latency; what you can deploy within fixed power and cooling budgets is also a critical factor.

Power efficiency shows how much useful storage service (in MB/s) you get per watt consumed, relating performance to power use. When we analyzed throughput per watt, a pattern emerged: Throughput per watt differed substantially (262x–435x throughput per watt) between the two building blocks, with the 1x SSD consistently higher across the tested mixes.<sup>9</sup>

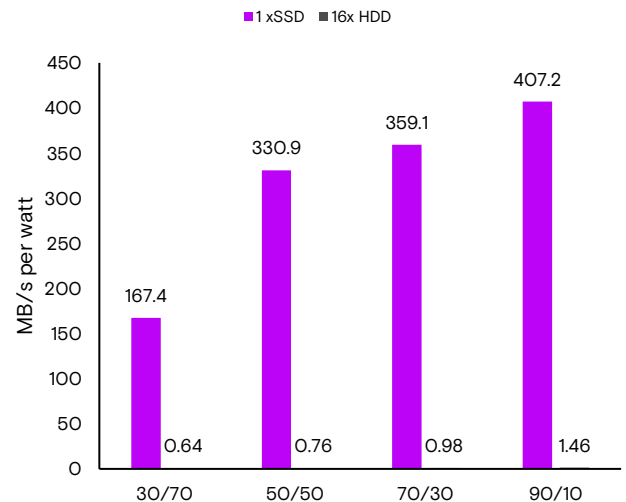


Figure 11: Storage power efficiency

7. See "Time to First Byte and Streaming Media" ([min.io](https://min.io)).

8. SSD advantage calculated using the measured values above as follows: (30/70): 180 / 5 = 36; (50/50): 196 / 3 = 65; (70/30): 227 / 3 = 76; (90/10): 289 / 3 = 96.

9. SSD advantages (MB/s per watt); 1x SSD watts were measured at the SSD. 16x HDD watts include the HDDs and the JBOD enclosure (necessary for use in these samples). Advantage calculated using the measured values above as follows: (30/70): 167.40 / 0.64 = 262; (50/50): 330.90 / 0.76 = 435; (70/30): 359.10 / 0.96 = 366; (90/10): 407.20 / 1.46 = 279.

## Throughput per TB

In object storage, raw throughput (MB/s) can mislead because it ignores potential dependency on deployed capacity; MB/s per TB shows the performance density you’re building. In HDD-heavy designs, larger drives add capacity faster than per-TB bandwidth, revealing a diminishing-density effect.

Figure 12 shows that the 1x SSD configuration delivered higher throughput per TB in all four workloads, and that 90/10 produced the highest performance density. In contrast, the 16x HDD configuration showed low performance density across all workloads, with performance per TB remaining below 1 MB/s.

1x SSD delivers 47x to 58x usable performance per unit of capacity across the full workload set, despite having comparable raw capacity (245TB SSD versus 256TB HDD array).<sup>10</sup>

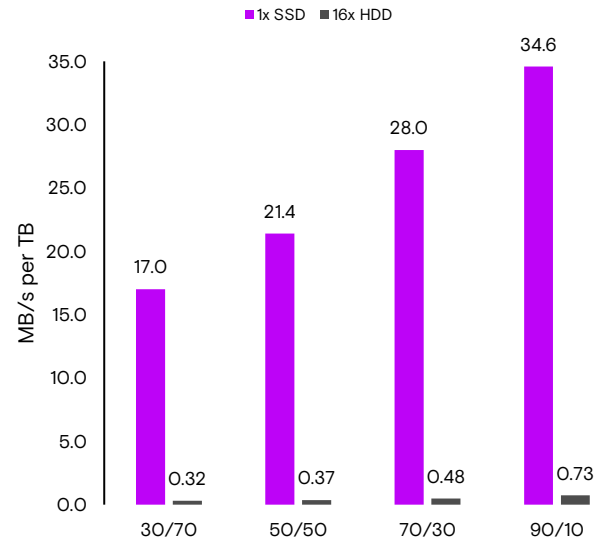


Figure 12: Throughput per TB

## Conclusion

When every watt and every rack unit count, storage efficiency becomes a key design variable.

Using the same MinIO Warp workload mixes, this evaluation shows how SSD- and HDD-based object-storage building blocks differ in throughput, tail latency/TTFB, throughput per watt, and throughput per TB—results that can help architects select media that fits the performance, responsiveness, and efficiency targets.

In power-, cooling-, and space-limited environments, higher throughput per watt translates into more work per rack and potentially lower operating cost, while lower tail latency and faster time to first byte can help reduce pipeline stalls and retries in AI training and analytics.

Together, performance density (MB/s per TB), efficiency (MB/s per watt), and responsiveness (latency/TTFB) define decision framework, enabling choices that prioritize capacity, cost, pipeline speed, and balance within rack and power constraints.

10. SSD advantage (MB/s per TB) calculated using the measured values as follows: (30/70): 17.0 / 0.32 ≈ 53; (50/50): 21.4 / 0.37 ≈ 58; (70/30): 28.0 / 0.48 ≈ 58; (90/10): 34.6 / 0.73 ≈ 47.

[micron.com/6600-ION](https://micron.com/6600-ION)