

Benchmark Summary

Aerospike vs. Cassandra - a Brief Comparison

As a leading NoSQL database, Aerospike delivers fast, predictable performance at scale at a much lower Total Cost of Ownership (TCO) than other alternatives. Our tests with Apache Cassandra showed that Aerospike delivered 15.4x better transaction throughput, 14.8x better 99th percentile read latency and 12.8x better 99th percentile update latency. And Aerospike did that at less than 10% of Cassandra's cost.

Perhaps that's hard to believe. Why not explore the details yourself?

Benchmark Summary

We tested Aerospike 4.6.0.4 Community Edition and Apache Cassandra 3.11.4 using the Yahoo Cloud Serving Benchmark (YCSB). To run a balanced mix of read and write operations for 24 hours on 2.65 TB of unique user data with a replication factor of 2, the total user data was 5.3 TB. Our environment included three Dell® R730xd rack-mounted servers with 128 GB of DRAM and two 14 core Xeon processors (56 hyperthreads) on each server. Each server utilized two 1.6 TB Micron 9200 Max NVMe SSDs for data and ran Centos Linux version 7.6.

As Table 1 presents Aerospike's throughput and latency advantages at the 99th percentile and 95th percentile, which were unmatched.

	Read Results			Update Results		
	Throughput Transactions (per second)	95th Percentile Latency (µsecs)	99th Percentile Latency (µsecs)	Throughput Transactions (per second)	95th Percentile Latency (µsecs)	99th Percentile Latency (µsecs)
Apache Cassandra 3.11.4	23,999	8,487	17,187	23,999	8,319	20,109
Aerospike CE 4.6.0.4	369,998	744	1,163	369,998	1,108	1,574
Aerospike advantage	15.4x better	11.4x better	14.8x better	15.4x better	7.51x better	12.8x better

Table 1: YCSB results summary

Maybe you're wondering why Cassandra didn't fare better. We followed various Cassandra tuning guidelines because we realized that its performance could be gated by caching, garbage collection, compactions and other factors. Our efforts yielded improvements, but ultimately Cassandra couldn't come close to the read/write speeds or throughput of Aerospike, as Table 1 indicates.

It's worth noting that we had a tough time populating Cassandra with the 2 billion objects needed for our tests. Initially, we ran YCSB clients without throttling. That yielded an insert rate of 83K Operations Per Second (OPS) but caused some data to be dropped -- something we discovered only after running read tests. We tried different insert rates to avoid data loss. At 40K OPS, we successfully loaded all of the data. But Cassandra took 14 hours to complete the job, and each of our nodes had more than 4000 pending compactions. This forced us to wait 5 days for the compactions to complete before we could initiate read/write benchmark tests. In contrast, we loaded all keys into Aerospike in less than 1 hour, and the database was ready immediately.

TCO Savings

Aerospike's leadership in transaction throughput and runtime performance is most remarkable when considered in the context of TCO. As mentioned earlier, Aerospike delivered its results at less than 10% of Cassandra's overall costs. That's worth a closer look.

To realistically compare TCO, we used Aerospike's results on a 3-node cluster as the target SLA and tried to size Cassandra's environment to deliver the same throughput and latencies. But even with different caching assumptions and hit ratios, Cassandra couldn't achieve the desired SLA for 99th percentile reads and writes because of garbage collection issues. We settled on a 95% page cache hit ratio for Cassandra, which yielded latencies closest to — yet still higher than — Aerospike's 95th percentile latencies.

If we ran one Cassandra instance on each node, we'd need 73 servers to approach the target SLA achieved by Aerospike on a 3-server cluster. Since many Cassandra users run multiple instances on a single server (through virtualization or containers), we assigned 3 instances per server, lowering Cassandra's size to 25 servers.

The differences in initial hardware costs were substantial. As Table 2 illustrates, the Aerospike cluster cost 8% of the Cassandra cluster.

	Cassandra (3 instances/server)			Aerospike (1 instance/server)			% cost savings with Aerospike
	Quantity	Cost	Subtotal	Quantity	Cost	Subtotal	
Servers (Dell R730xd)	25	\$10,842	\$271,050	3	\$10,842	\$32,526	
Memory (32GB DIMM)	292	\$671	\$195,932	4	\$671	\$2,684	
Storage (Micron 9200)	50	\$1,053	\$52,650	6	\$1,053	\$6,318	
Cassandra Total			\$519,632	Aerospike Total		\$41,528	92%

Table 2: Initial hardware cost for target SLA

But TCO involves more than hardware costs. Table 3 details hardware, power, cooling and support costs over a three-year period for a sample production scenario. Using the capacity and performance data from our benchmark, we increased the data size from 2 billion unique keys to 20 billion for Year 1. This yielded 18 TB of unique data and 36 TB of replicated data. For Year 2, we modeled a high-growth business scenario that drove the number of unique keys to 30 billion and increased the cluster size by 50%. Year 3 brought more growth; the number of unique keys reached 39.9 billion, increasing the cluster size by 33% over the previous year.

So what happened? As Table 3 shows, Aerospike delivered a 90% cost reduction over Cassandra, saving nearly \$9.27 million for the three-year period.

	Cassandra			Total	Aerospike			Total
	Year 1	Year 2	Year 3		Year 1	Year 2	Year 3	
# of servers	244	365	486		30	45	60	
Cost per server for 3 years	\$7,000				\$5,211			
Servers	\$1,708,000	\$2,555,000	\$3,402,000	\$7,665,00	\$156,330	\$234,495	\$312,660	\$703,485
Power	\$170,567	\$255,152	\$339,737	\$765,456	\$20,971	\$31,457	\$41,942	\$94,370
Cooling	236,899	\$354,378	\$471,856	\$1,063,133	\$29,127	\$43,690	\$58,252	\$131,069
Space	\$6,000	\$6,000	\$6,000	\$18,000	\$770	\$770	\$770	\$2,310
Support	\$174,268	\$260,689	\$347,109	\$782,066	\$21,426	\$32,139	\$42,853	\$96,418
Total	\$2,295,734	\$3,431,219	\$4,566,702	\$10,293,655	\$228,624	\$342,551	\$456,477	\$1,027,652

Table 3: Comparative 3-year TCO scenario

Predictable Performance

In production environments, predictable performance is essential. That's why it's important to look beyond average read/write response times and compare the *ranges* of response times and throughput. After all, every business needs its infrastructure to perform consistently under peak load. As it turns out, our analysis showed that Aerospike delivers far greater performance predictability than Cassandra.

Let's start with the variances for reads. Aerospike serviced reads more quickly and with greater predictability than Cassandra, as Fig. 1 displays. Note how few spikes -- variances -- in read latencies were observed with Aerospike (in red) compared to Cassandra (in blue). Indeed, we had to use a log graph to display comparative data because Aerospike's read latencies were so low. Aerospike produced 740K OPS with a 99th percentile 1.2 ms latency while Cassandra produced 24K OPS at 99th percentile of 17.0 ms.

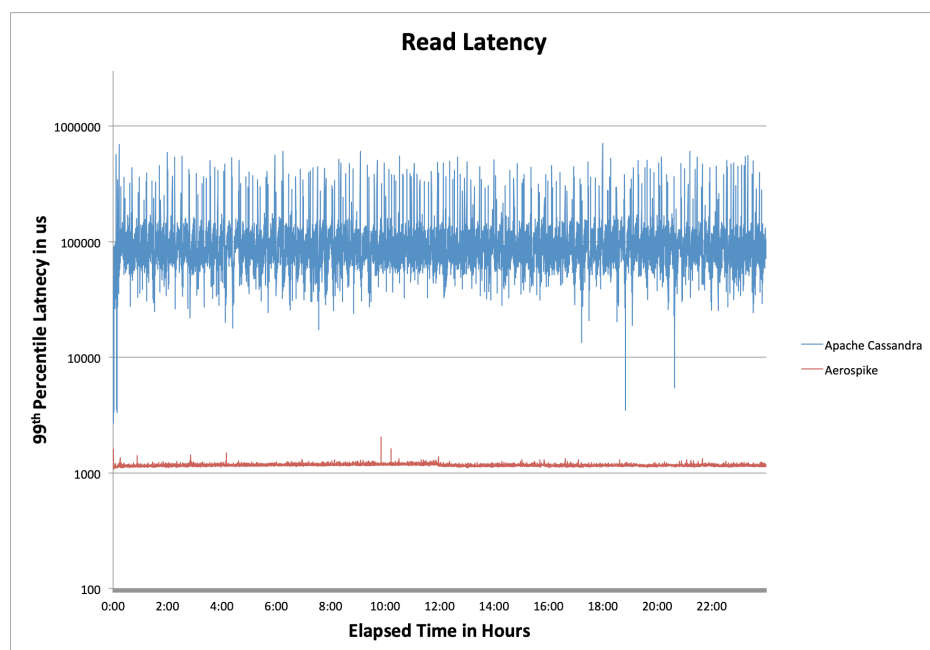


Fig. 1: Read latency variances observed for Cassandra and Aerospike

What about throughput? As Fig. 2 presents that Aerospike serviced more reads and did so with more consistent performance than Cassandra. Aerospike generated 369,998 OPS with little variance compared to Apache Cassandra's throughput of 23,999 OPS.

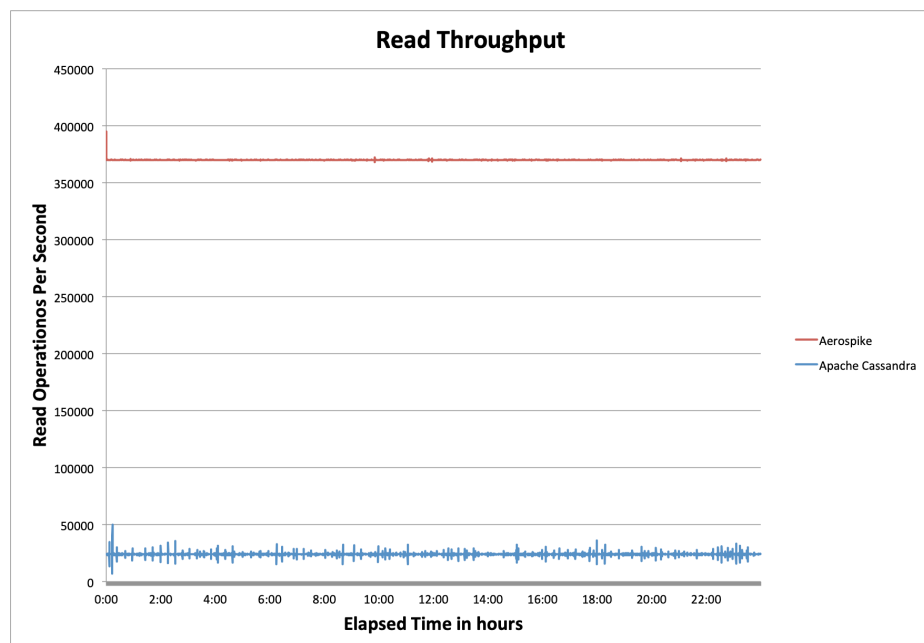


Fig. 2: Read throughput variances observed for Cassandra and Aerospike

If you're wondering about writes, the situation is similar. Fig. 3 illustrates how Aerospike processed writes far more quickly and with far more consistent performance than Cassandra.

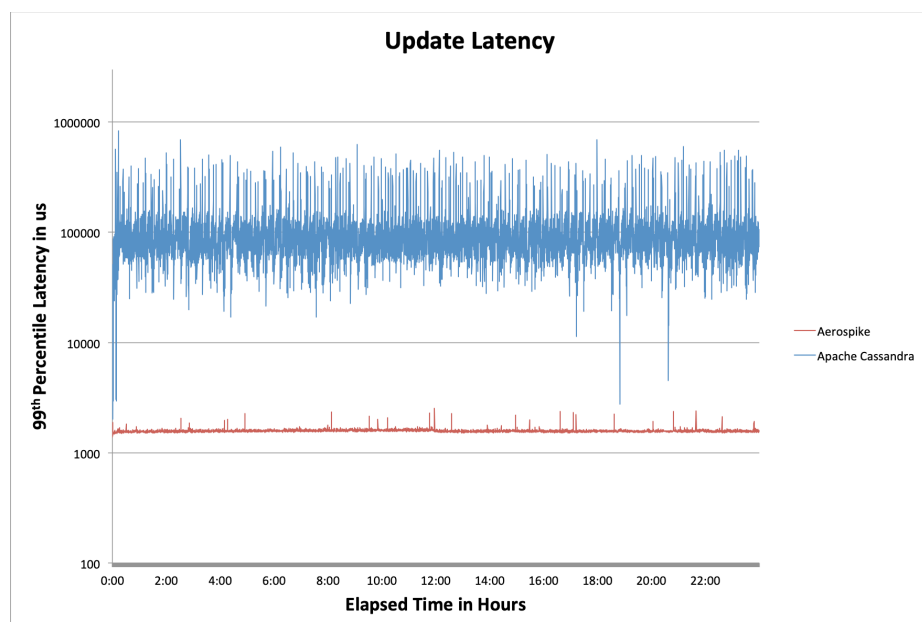


Fig. 3: Write latency variances observed for Cassandra and Aerospike

Furthermore, Aerospike's throughput was more predictable -- and much higher -- than Cassandra's, as shown in Fig. 4. Aerospike generated 369,998 OPS with 1.6 ms 99th percentile latency versus 23,999 OPS at a 99th percentile latency of 20.0 ms.

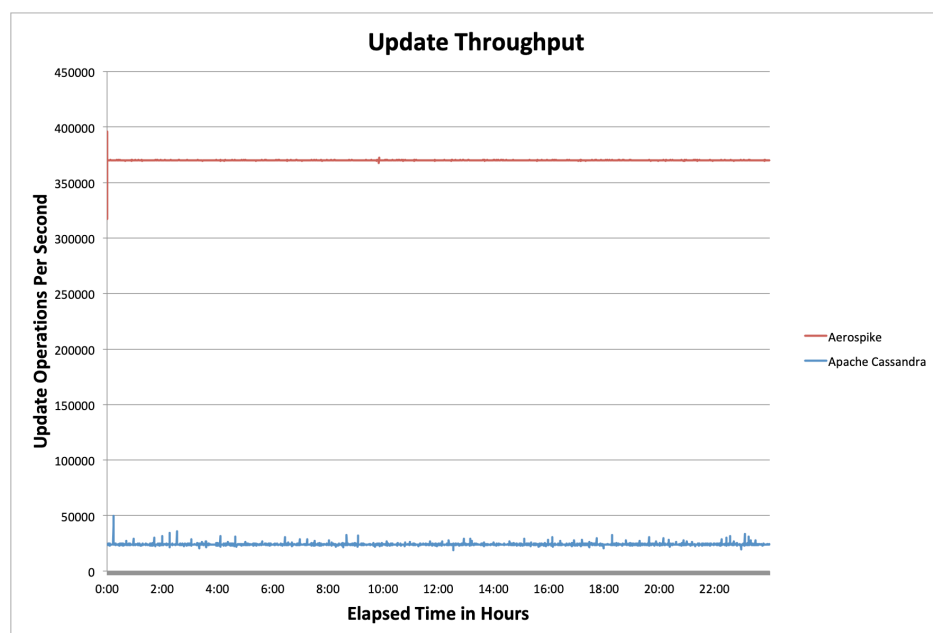


Fig. 4: Write throughput variances observed for Cassandra and Aerospike

What Does this Mean for You?

Our tests demonstrated that Aerospike delivers higher throughput with lower latency at a much lower cost than Cassandra; for more details, just download our [whitepaper](#). You'll see how Aerospike's predictable performance makes it easy for companies to meet their application SLAs even as business demands grow.

Ultra-fast, predictable performance at scale. Small server footprints. Operational ease. High availability and uptime. Low TCO. With Aerospike, you don't need to trade one for the other.

To learn more, contact Aerospike for a technical briefing. Discover why many Cassandra users have already migrated to Aerospike for their production applications, and explore what Aerospike can do for you.

About Aerospike

Aerospike is trusted by leading enterprises around the world to help them build and deploy modern data architecture solutions with confidence. The Aerospike enterprise-grade non-relational NoSQL database helps companies power mission-critical, strategic operational applications that make digital transformation possible. Powered by a patented Hybrid Memory Architecture™ and autonomic cluster management, Aerospike is used by enterprises in the financial services, banking, telecommunications, technology, retail, e-commerce, ad tech, martech and gaming industries, and is well-suited for fraud prevention, digital wallet, online brokerage, real-time analytics and other applications that require extreme uptime, performance and scalability. Aerospike customers include Adobe, Bharti Airtel, FlipKart, Kayak, Nielsen, and Snap. The company is headquartered in Mountain View, Calif.

©2019 Aerospike, Inc. All rights reserved. Aerospike and the Aerospike logo are trademarks or registered trademarks of Aerospike. All other names and trademarks are for identification purposes and are the property of their respective owners.