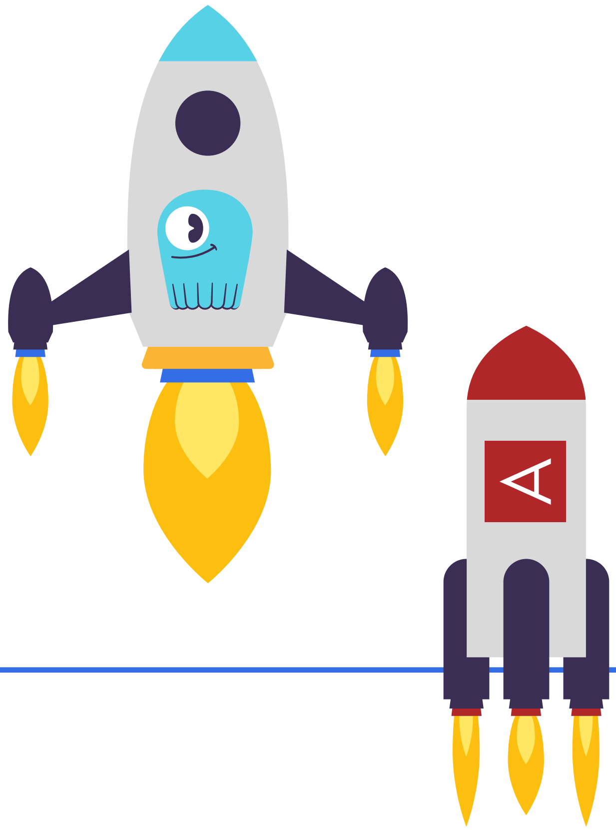
ScyllaDB vs. Aerospike: A Comparison



SUMMARY OF RESULTS

If you're in need of a high performance NoSQL database built for next generation applications there are a few leading candidates. In this comparison we'll show how ScyllaDB outperforms Aerospike — using Aerospike's preferred test method.

Aerospike generated a 20-node cluster of AWS EC2 i3en.24xlarge servers benchmark environment and report supporting 5 million reads per second against a 1 petabyte data set in a read-only workload test. Aerospike is a serious competitor, with a refined C engine that can beat a wide column database as Cassandra easily, primarily due to its novel, simplified key-value model and a key cache which stores all keys in RAM.

Us at ScyllaDB wondered whether we can beat Aerospike, in its own game, using a similar scale cluster of 20 i3en.metal servers (same physical machine), ScyllaDB was able to sustain 7 million reads per second — **a 40% improvement in throughput**.

As ScyllaDB is a more robust database in terms of data modeling, consistency options and scalability, this result is a great validation to our close-to-the-hardware design.

TEST RESULTS

ScyllaDB is capable of up to 40% greater throughput on equivalent hardware.

Workload	ScyllaDB	Aerospike	% Difference
1 PB User Data Writes (100%)	7.5 million per second	NA	NA
1 PB User Data Reads (100%)	7 million per second	5 million per second	**ScyllaDB +40%**
1 PB User Data Read/ Writes (80%/20%)	5 million per second	3.7 million per second	**ScyllaDB +35%**

THE PLAYERS

ScyllaDB is a next generation NoSQL database designed for data-intensive applications that require high throughput, low latency, and predictable scalability. Built with architectural advancements such as a close-to-the-metal architecture, sharding per core, and auto performance tuning, ScyllaDB supports wide-column and key value modeling for hundreds of gigabytes to petabytes of data and with high availability. It is also compatible with both the Apache Cassandra (CQL) and Amazon DynamoDB APIs, allowing ScyllaDB to integrate into the ecosystems of drivers, integrations, and tools that support these other databases for seamless migration.

Aerospike is positioned as a real-time NoSQL data platform with unlimited scale, predictable performance, and industry-leading uptime. Originally designed for key-value data models, it recently expanded positioning to multi-model and include JSON document database capabilities.

Both ScyllaDB and Aerospike databases are utilized across many of the same use cases such as Customer 360, fraud & threat detection, recommendation/personalization, AI/ML and IoT.

DIFFERENCES FROM THE CODE BASE ON UP

ScyllaDB uses a "shared nothing" distributed processing framework known as Seastar, and is written in C++, to take full advantage of modern multiprocessor servers that can have upwards of 100 or more CPU cores. It also operates with its own custom CPU and IO schedulers, so it can bypass the kernel and thus avoid system interrupts and blocking operations. It has no hard limit to the number of nodes it can have in a cluster.

Aerospike is written in C, and also uses a "shared nothing" distributed processing approach.

Both ScyllaDB and Aerospike are leaderless peer-to-peer active-active clustering topologies.

COMPARISON OF OFFERINGS

ScyllaDB is available as free ScyllaDB Open Source, as commercial ScyllaDB Enterprise for self-managed deployments, and as ScyllaDB Cloud that can run as a fully managed Database-as-a-Service (DBaaS) on AWS and Google Cloud. [See here]

Aerospike is available as a free, non-open-source Community Edition, as a commercial Aerospike Server Standard Edition and Enterprise Edition, and as a Managed Cloud Service on AWS and Google Cloud. [See here]

ScyllaDB Open Source vs. Aerospike Community Edition

- ScyllaDB Open Source is a fully-functional open source software (AGPL licensed) offering that can be used without restriction in terms of namespaces, rows, nodes per cluster, or total size.
- Aerospike Community Edition is a commercially-licensed product artificially limited to only supporting two (2) namespaces, four billion objects per namespace, eight (8) nodes per cluster, and 5 terabytes of data total.

ScyllaDB Enterprise vs. Aerospike Enterprise Edition

- ScyllaDB Enterprise is a fully-functional commercially-licensed product offering that has the technical capability to scale to any number of nodes, any number of namespaces, and any number of rows.
- Aerospike Enterprise Edition a commercially-licensed product that is hard limited to 256 nodes per cluster, 32 namespaces, and 500 billion objects per namespace.

ScyllaDB Cloud vs. Aerospike Cloud Managed Service

- ScyllaDB Cloud is available for use on both AWS and Google Cloud; ScyllaDB provides an open cost calculator for sizing and pricing estimation prior to purchase. It can be purchased directly from ScyllaDB or via the AWS or Google Cloud Marketplace.
- Aerospike Cloud is provided in early availability but not in GA.

Which means that any node can accept write, update, or delete operations. (Compare this to other databases that have one primary replica that can accept writes, and others are read-only replicas.)

LIMITLESS VS. LIMITED SCALABILITY

ScyllaDB has no artificial limits to the number of server nodes in the cluster, the number of terabytes of total storage, nor the number of namespaces you can have in a cluster. Users run up to thousands of namespaces per cluster.

Aerospike is limited to a maximum of 256 nodes in a cluster for its Enterprise versions. Aerospike's Community Edition is further limited to only 8 nodes and a total of only 5 terabytes of data. Aerospike also limits the number of namespaces per cluster to 32 on its Enterprise edition, and only supports 2 namespaces per cluster on its Community Edition.

DATA MODEL

ScyllaDB is a wide column NoSQL database, often described as a "key-key-value" database. It uses a partition key to distribute data around a cluster, evenly and efficiently, and a clustering key to sort data rows within partitions.

Aerospike is a key-value database. While it also supports even data distribution using its partitioning algorithm, it does not have clustering keys.

QUERY LANGUAGE

ScyllaDB primarily uses the Cassandra Query Language (CQL), which it shares with Apache Cassandra, DataStax Enterprise, Amazon Keyspaces, Microsoft Azure Cosmos DB, and Yugabyte. While it is not a formal standard such as ANSI/ISO SQL, CQL has become a de facto industry standard. This means ScyllaDB is compatible with a broad ecosystem of open source drivers, connectors, code and applications that support CQL-based systems.

ScyllaDB is also compatible with the Amazon DynamoDB API. It thus shares the same JSON-format query language, and the same SDKs, tools and services available for DynamoDB.

Aerospike uses a proprietary Aerospike Query Language (AQL). It cannot leverage a broader ecosystem other than that which Aerospike or its close partners produce themselves.

GLOBAL DEPLOYMENT AND CONSISTENCY

Both ScyllaDB and Aerospike have methods to replicate data globally.

ScyllaDB's multi-datacenter replication allows a single logical cluster to span multiple locations / availability zones. It uses internal mechanisms to determine which, if any, data is replicated to other data centers. ScyllaDB allows for tunable consistency, allowing users to determine how they want to manage consistency per operation, with options such as all (requiring all nodes in a global cluster to achieve the same value), quorum (a majority of global nodes in the cluster), local quorum (a majority of local nodes in the coordinator node's data center), and so on. This gives users great flexibility in managing their data and cluster deployments.

Aerospike offers two different methods to do global data replication. The first, Cross-Datacenter Replication (XDR), allows data to be replicated across datacenters, but the clusters themselves are treated as separate logical units, operating independently. This means it cannot guarantee data consistency across clusters.

The second, multi-site clustering, is more akin to ScyllaDB, allowing the different local deployments to act as a single logical cluster. But it requires strong consistency across the globally deployed database, which could lead to long delays or timeouts. While Aerospike does have flexibility of replication options, it lacks granular, per operation controls like ScyllaDB's tunable consistency.

STORAGE

ScyllaDB prefers to run on fast NVMe SSDs, optimized and striped as per RAID 0. On such a storage subsystem, ScyllaDB uses a custom I/O scheduler that minimizes latency and allow fair access to the disks by hundreds of cores. ScyllaDB writes directly to the disk from user space. Data is stored in immutable Log Structured Merge (LSM) tree files in a format known as Sorted Strings Tables (SSTables). This storage format is fully compatible with Apache Cassandra's SSTable format. By using the Linux filesystem, normal administrative tools can be used to manage the files. ScyllaDB also comes with a utility that automatically determines the optimal level of parallelism at which it can send read and write requests to disk, and the I/O scheduler will automatically manage that during periods of peak workload to achieve the optimal I/O throughput and latency.

Aerospike uses custom SSD drivers to write directly to the storage layer (direct I/O). These drivers require users to manually partition larger physical disks into 2 TiB partitions. Users have to make manual choices during time of installation as to how to partition their drives for optimal performance. Further, because Aerospike writes directly to disk, normal drive and filesystem management applications cannot be used.

CONSENSUS ALGORITHMS

ScyllaDB uses a Paxos consensus algorithm for strict linearizability of writes, known as compare-and-set (CAS) operations or Lightweight Transactions (LWT). However, Paxos is not used across all operations to avoid performance-impacting delays on all operations; LWTs are applied as desired by the user application to certain tables and write queries. Instead, generally ScyllaDB uses a tunable consistency model per operation where users can control how strictly or loosely they wish to control database operations.

ScyllaDB also includes an implementation of the Raft consensus algorithm to keep schema changes immediately consistent across the cluster.

Aerospike uses Paxos for linearizability and strong consistency of single record operations. Aerospike offers eventual consistency operations as well.

PERFORMANCE COMPARISON

Aerospike published a petabyte-scale operational workload whitepaper in August 2021, stating that they could support such a workload on 20 AWS EC2 i3en.24xlarge nodes. In this test, Aerospike claimed to be able to support 5 million transactions per second (TPS) for a read-only workload and 3.7 million TPS for a mixed read-write workload (80% reads and 20% writes, at rates of 3 million reads per second and 750,000 writes per second). Readers will note there was a 27% lower aggregate performance for the mixed workload, even for 20% writes.

ScyllaDB later replicated the conditions of this test on nearly identical hardware — using 20 AWS EC2 i3en.metal servers. These had the same number and kind of CPUs, the same amount of RAM and the same amount of storage. Yet ScyllaDB was able to sustain ***7 million operations per second for reads or writes***, with single-digit millisecond latencies. In fact, ScyllaDB was able to sustain writes at ***7.5 million*** inserts per second with 4 ms P99 latencies. So while Aerospike was able to maintain ≤1ms P99 latencies, it provided significantly lower throughput than ScyllaDB.

ScyllaDB offered throughputs 40% to 100% greater than Aerospike on equivalent hardware, depending on the read/write workload. ScyllaDB achieved slightly slower latency for P99 reads; we believe that had we run the same throughout as Aerospike then ScyllaDB latency would have been equivalent.

TESTING TWO WORKLOADS

Aerospike's test condition was actually broken into two separate workloads running on the same cluster. The first was a simulated user profile namespace comprising 500 billion unique keys, for a unique uncompressed, unreplicated size of 1 petabyte. This workload used an "AP" consistency mode ("Available, Partition Tolerant" as per the CAP theorem). On this workload, Aerospike applied an 80% read, 20% write workload. Note that Aerospike cannot store more than 500 billion keys per namespace.

The second dataset was a marketing campaign namespace, composed of 6 billion unique keys, for 1.5 TB of primary, uncompressed, unreplicated data. This workload used a "CP" consistency mode ("Consistent, Partition Tolerant" as per the CAP theorem). To this was applied a mixed 50% read, 50% write workload.

REPLICATION

Also note Aerospike's setup for the test conditions used a replication factor of two. ScyllaDB supports multiple replication factors, on a per-datacenter and keyspace granularity. Typically, ScyllaDB suggests a replication factor of 3 to ensure data remains highly available and the cluster is more resilient in case of any node outages.

COMPRESSION

Aerospike's petabyte of raw data was compressed by a factor of 4, from a base of 1 TB to 250 TB. Then it was replicated by a factor of two for resiliency, to a total of 500 TB of data stored on the cluster. This is well-within the 1,200 TB total data storage available in a cluster of 20x i3en.24xlarge servers.

For ScyllaDB, we used a compression factor of 3x, to 333.3 TB of data, then replicated, for a total of 666.6 TB. Also well within the 1,200 TB total data storage available.

WORKLOAD PRIORITIZATION

The Aerospike benchmark further divided its workload into two components: one was to simulate a user profile store, and the second to represent a campaign. Generally, this results in smaller workloads getting starved for system resources. Aerospike proved successfully that their system has enough capacity and fairness to handle the two workloads.

ScyllaDB provides a finer level of control. A user can associate a workload with a dynamic amount of shares. When the cluster isn't loaded, every workload can utilize the entire cluster resources. However, when the system utilization is high, priorities are enforced and the system guarantees a better SLA to the workloads with higher shares. This feature is known as workload prioritization and is unique to ScyllaDB, utilizing its CPU and IO schedulers.

IMPACT OF WORKLOAD PRIORITIZATION

Using this feature, it was presumed the smaller campaign workload was more important because of its higher mix of writes in the workload. Thus, by deprioritizing the user profile we were able to improve the performance of the campaign workload significantly — getting P99 write latency to less than half. However, this did commensurately increase latencies on the user profile workload. This fits well with a data analytics workload that requires throughput and is not latency sensitive. Users need to apply workload prioritization judiciously understanding the control and tradeoffs it provides.

TOTAL COST OF OWNERSHIP

Aerospike stated in its published benchmark that it could save users running Apache Cassandra significant costs. They cited a one year total cost of ownership (TCO) for a petabyte-scale workload on Cassandra as $4 million, and that Aerospike would be able to do the same workload for $1.23 million annually on a 20 node cluster.

ScyllaDB was able to test the same workloads on equivalent hardware and observed it provided between ***40% to 100% greater throughput than Aereospike*** (depending on the workload). Extrapolating from this, yet understanding that there is still the need to store a petabyte of raw information, ScyllaDB should be able to run the same workload on a scaled down 13 node cluster for $800,000 annually. This would save about a third of Aerospike's projected annual costs.

Alternately, users could provision a cluster at the exact same size and equivalent cost as the one proposed by Aerospike, yet get far greater throughputs with ScyllaDB.

Workload (in TPS)	Campaign: 200k R/W		User profile: 5M 80/20 R/W	
	before: 1000 shares	after: 1000 shares	before: 1000 shares	after: 500 shares
Write latency (in ms)	0.632 P50 2.454 P99	0.354 P50 1.184 P99	0.326 P50 1.252 P99	0.440 P50 3.244 P99
Read latency (in ms)	1.195 P50 4.555 P99	0.855 P50 3.731 P99	0.744 P50 3.709 P99	1.043 P50 6.455 P99

SOURCES

Benchmarking Petabyte-Scale NoSQL Workloads with ScyllaDB [July 2022]

Operating at Monstrous New Scales: Benchmarking Petabyte Workloads on ScyllaDB [Feb 2022]

Running Operational Workloads with Aerospike at Petabyte Scale in the Cloud on 20 Nodes [August 2021]

ABOUT SCYLLADB

ScyllaDB is the database for data-intensive apps that require high performance and low latency. It enables teams to harness the ever-increasing computing power of modern infrastructures – eliminating barriers to scale as data grows. Unlike any other database, ScyllaDB is built with deep architectural advancements that enable exceptional end-user experiences at radically lower costs. Over 400 game-changing companies like Disney+ Hotstar, Expedia, FireEye, Discord, Crypto.com, Zillow, Starbucks, Comcast, and Samsung use ScyllaDB for their toughest database challenges. ScyllaDB is available as free open source software, a fully-supported enterprise product, and a fully managed service on multiple cloud providers. For more information: ScyllaDB.com

United States Headquarters
2445 Faber Place, Suite 200
Palo Alto, CA 94303 U.S.A.
Email: info@scylladb.com

Israel Headquarters
11 Galgalei Haplada
Herzelia, Israel