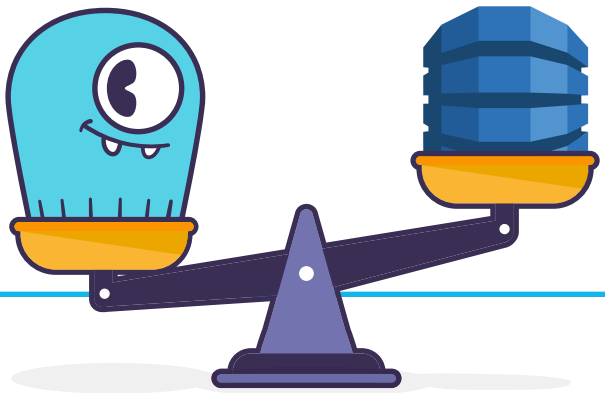
SCYLLA

Scylla vs. Amazon DynamoDB

Evaluating NoSQL Databases for Performance and Cost



CONTENTS

OVERVIEW

Over the past decade, NoSQL databases have carved out a place as the standard data platform for modern applications that use unstructured data at scale. These applications have recently exploded in popularity; unstructured data makes up 90% of all corporate data, growing 50 times faster than traditional structured data.

For this reason, NoSQL databases have found applications across industries. Within enterprises, unstructured data often has multiple uses. It's valuable for both high speed operational workloads and for 'Big Data' analytics. Use cases that benefit significantly from NoSQL capabilities include:

- Internet of Things, network monitoring, and time series
- Profile management and customer 360
- Security monitoring and fraud detection
- Product catalogs and shopping carts
- AdTech and real-time bidding

The sheer number of NoSQL databases on the market today makes it difficult to perform detailed comparisons among available offerings. In this document, we provide IT leaders with a basis for comparing two leading NoSQL databases: Amazon DynamoDB and Scylla NoSQL database.

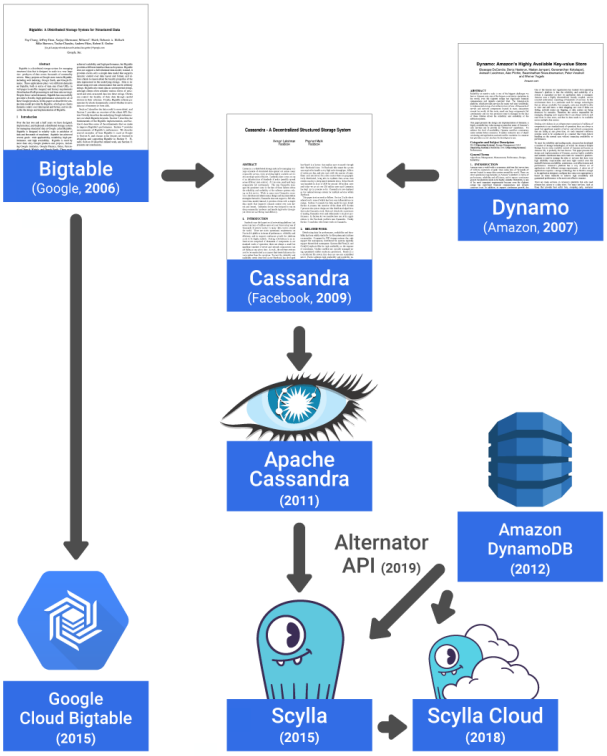
BACKGROUND

Scylla and DynamoDB are on the same evolutionary branch of the NoSQL family: highly scalable, highly distributed databases. Ideas from the original Google Bigtable and Amazon Dynamo white papers, which ushered in the era of NoSQL, helped influence the original implementation of Cassandra at Facebook. Cassandra was thereafter made an open source project of the nonprofit Apache Foundation.

While the original Dynamo was used exclusively as an internal database at Amazon, its commercialized successor, DynamoDB, was released based on the same principles as the original Dynamo paper.

Scylla entered the market as a reimplementation of Cassandra, re-written from the ground up in C++ and re-architected to take advantage of modern multicore servers using highly asynchronous communications.

After having achieved feature parity with Cassandra, Scylla took the evolutionary step to add DynamoDB API compatibility, dubbed Project Alternator. Now with Scylla users could run their DynamoDB workloads anywhere — on another public cloud, or on-premises.



This image shows the "family tree" of highly scalable NoSQL databases most closely related to Scylla. Many of the key concepts found in the original white papers can be found in the current implementation of Scylla.

As such, Scylla and DynamoDB provide an excellent case for comparison. While they share the same technical heritage, Scylla and DynamoDB diverge significantly in practice. The differences are best demonstrated through industry-standard performance benchmarking. Our goal in this paper is to provide a concrete, empirical basis for selecting Scylla over DynamoDB.

In this document, we compare Scylla with Amazon DynamoDB. The high-level takeaway of this study is this: Scylla performs significantly

better than Amazon DynamoDB under real-world conditions. It also delivers significant cost savings over Amazon DynamoDB.

CHOOSING THE RIGHT DATABASE: THE BUSINESS VALUE

While a database is an infrastructural component that is rarely, if ever, directly exposed to your customers, it is the foundation upon which modern businesses are built. A solid choice can help anchor organizational acceleration and growth. But a choice that is made with only short-term thinking in mind can lead to severe long-term consequences.

DynamoDB is a very facile database to begin development with. It is always there as an option for users of AWS, and many of them drift towards it for rapid prototyping and getting to production quickly.

With AWS' entire infrastructure capabilities behind them, scalability is essentially near infinite. However, there are a few issues with this. First is the problem that comes when you look at your bill at the end of the month. As your use of DynamoDB grows, so does your monthly bill. While those costs may be acceptable during low-volume early-adoption phases, many organizations find themselves trapped once operating at scale.

And this is where the second issue comes into play. Before ScyllaDB introduced its DynamoDB compatible API, DynamoDB users could run their workloads only on AWS. They were locked in to a single vendor. If they wanted to explore other options for affordability, they would need to remodel their data, change their queries to migrate their data off of DynamoDB to any other database solution.

This is where Scylla can offer operational flexibility. Users can negotiate with different public or private cloud providers to find the most affordable solution for them. They can even bring their workloads in house through an on-premises deployment.

Scylla has been tested on all major cloud providers, opening the opportunity to run multi-cloud and hybrid topologies. In practice, a major Scylla customer in telecommunications runs Scylla in their own data centers (private cloud) as well as on AWS instances, simultaneously. This optionality in itself provides a major benefit over DynamoDB.

SUMMARY OF BENCHMARK RESULTS

A high-level rollup of results is provided below. The test is a widely-used benchmark known as the Yahoo! Cloud Serving Benchmark (YCSB), which is considered the open standard for comparative performance evaluation of data stores. YCSB was developed at Yahoo! Labs to provide a framework and common set of workloads for evaluating the performance of different key-value stores

The benchmark defines a Service Level Agreement (SLA) of 120,000 operations per second, split evenly between reads and writes, with latency less than 10ms in the 99% percentile. Each database is provisioned with the minimum resources necessary to meet this SLA. Each database is populated first with 1 billion rows using the default, 10-column schema defined by the YCSB.

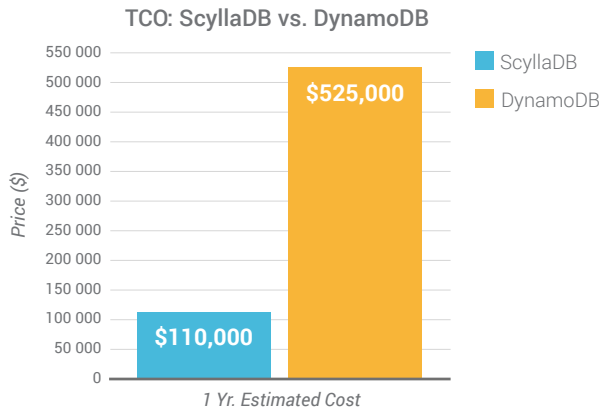
PERFORMANCE COMPARISON

The Yahoo! Cloud Serving Benchmark revealed the following:

- DynamoDB missed the required SLA multiple times, especially during the population phase.
- DynamoDB has 3x-4x the latency of Scylla, even under conditions favorable to DynamoDB
- DynamoDB is 7x more expensive than Scylla
- Dynamo was extremely inefficient in a real-life (Zipfian) distribution, requiring 3x capacity and 20x higher costs than Scylla
- Scylla demonstrated up to 20x better throughput in the hot-partition test with lower latency numbers

COST COMPARISON

Furthermore, the test demonstrates that Scylla delivers these performance gains while significantly reducing the total cost of ownership, even when running Scylla on AWS instances. The cost savings are summarized below.



Scylla Enterprise (3 x i3.8xlarge + Scylla Enterprise license)	**Amazon DynamoDB** (160K write | 80K Read + Business-level Support)
Year-term Estimated Cost: ~$71K	Year-term Estimated Cost: ~$524K
Assumptions	
i3.8xlarge cost: $42,000 (1-year contract, all upfront payment)	DynamoDB 1-year term: ~$288K
Scylla Enterprise License: $28.8K /per year (Total of 48 cores)	Monthly fee : ~$19.7K/ month (~$236K annual)

An added consideration, which is not included in this TCO calculation, is the cost associated with operational overhead.the human intervention required to monitor and maintain database infrastructure. Scylla requires only 3 medium powered instances to meet the YCSB SLA. When compared with similar NoSQL databases (notably Apache Cassandra), this Scylla cluster is small and therefore more easily managed and maintained than larger clusters of smaller instances.

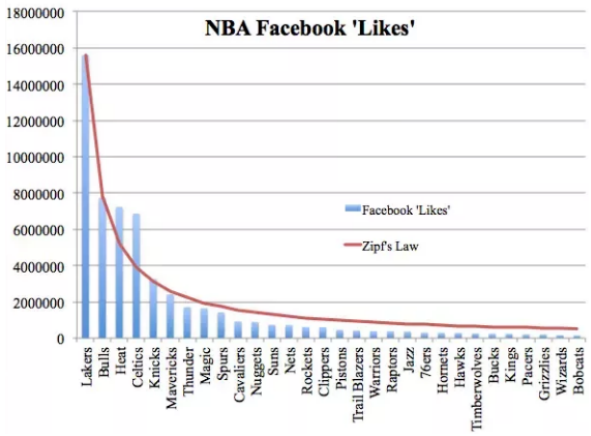
(For organizations looking to completely offload management, Scylla Cloud provides a fully managed version of Scylla. Depending on the plan selected, Scylla Cloud is 4-6x less expensive than DynamoDB.)

TESTING AGAINST REAL-WORLD DATA DISTRIBUTION

Ideally you design data schemas to produce a uniform distribution of primary keys. In practice, however, some keys are accessed more than others — "hot keys" — resulting in a situation referred to as "Zipfian Distribution." For example, it's common practice to rely on a UUID to query customers, or product IDs to query the product catalog, and then to retrieve the profile. Some customers are naturally more active than others, some products will be more popular than others, and in many cases a viral product can skew distributions suddenly.

Thus real-world distributions are often unpredictable, and the differential in access times can vary by up to 1000%. Developers are usually not in a place to improve the situation. Adding an additional column to the primary key to make the distribution more granular can improve the specific access, but at the cost of complexity once the full customer or product profile is retrieved.

Typical datasets often exhibit Zipfian distribution. In essence, a Zipfian distribution reflects the 20/80 power law; 20% of keys account usually for 80% of queries.



A real-world data set that displays Zipfian Distribution, a common, real-world pattern among datasets that affects database performance

YCSB Workload	Scylla 2.1 (3x i3.8xlarge)	DynamoDB (160K WR | 80K RD)
Workload A 50% Read / 50% Write Range: 1B partitions Distribution: Zipfian Duration: 90 minutes Hot set: 10K partitions Hot set access: 90%	Overall Throughput (ops/sec): **120.2K** Avg Load (scylla-server): ~55%	Overall Throughput(ops/sec): **65K** Avg Load (scylla-server): ~WR 42% | RD 42%
	READ operations (Avg): **~40.56M** Avg. 95th Percentile Latency (ms): 6.1	READ operations (Avg): **~21.95M** Avg. 95th Percentile Latency (ms): 6.0
	Avg. 99th Percentile Latency (ms): 8.6	Avg. 99th Percentile Latency (ms): 9.2
	UPDATE operations (Avg): **~40.56M** Avg. 95th Percentile Latency (ms): 4.4 Avg. 99th Percentile Latency (ms): 6.6	UPDATE operations (Avg): **~21.95M** Avg. 95th Percentile Latency (ms): 7.3 Avg. 99th Percentile Latency (ms): 10.8

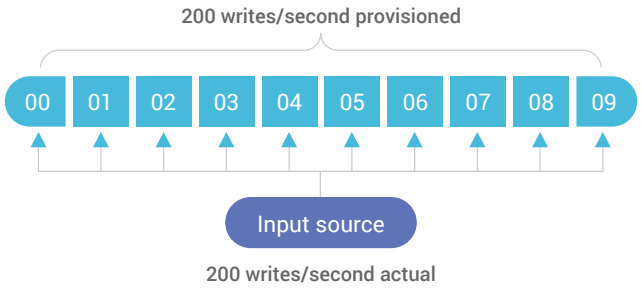
Testing against a dataset with Zipfian distribution reveals a performance limitation of DynamoDB. In the results below, note that Scylla supports a throughput roughly double that of DynamoDB: 120,000 operations per second, versus 65,000 for DynamoDB. Both read and update queries also average double the volume on Scylla versus DynamoDB.

The takeaway from these results is that normal data distributions require over-provisioning capacity on DynamoDB to approach the levels available with Scylla. Why can't DynamoDB meet the SLA in this case? The answer lies within the DynamoDB model. Global reservation is divided into partitions, each of which is limited to 10GB.

$$\text{\# of Partitions}_{\text{(For throughput)}} = \frac{\text{RCU}_{\text{for reads}}}{300\ \text{RCU}} + \frac{\text{WCU}_{\text{for writes}}}{1000\ \text{WCU}}$$

$$\text{\# of Partitions}_{\text{(For size)}} = \frac{\text{Table Size in GB}}{10\ \text{GB}}$$

$$\text{\# of Partitions} = \text{MAX}\left(\text{\#of Partitions}_{\text{(For size)}} \mid \text{\#of Partitions}_{\text{(For throughput)}}\right)$$

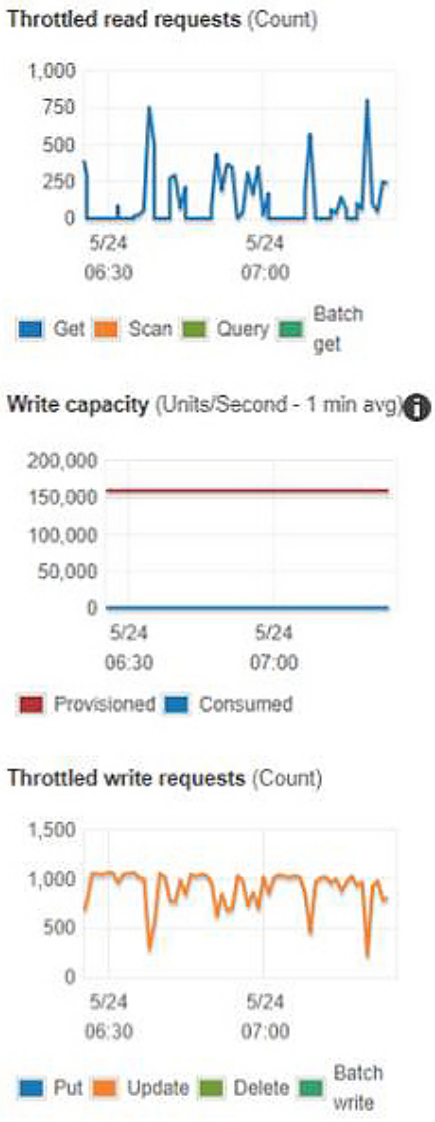


Here is the problem: a partition accessed this way can hit its throttling cap even when overall traffic is within the global reservation. In the example above, when reserving 200 writes, each of the 10 partitions cannot be queried more than 20 writes per second.

TESTING AGAINST A HOT PARTITION

To explore this 'hot partition' issue in greater detail, we ran a single YCSB benchmark against a single partition on a 110MB dataset with 100K partitions. The test exposed a DynamoDB limitation when a specific partition key exceeded 3000 read capacity units (RCU) and/or 1000 write capacity units (WCU).

Even when using only ~0.6% of the provisioned capacity (857 operations per second), DynamoDB throttled requests, returning `ProvisionedThroughputExceededException` (otherwise known as Code 400) errors. This throttling can be seen in the metrics below:



Though the system was provisioned for 75,000 writes per second, and 150,000 reads per second, in reality, it could achieve only brief peaks of performance at 750 reads per second (with the max, as stated, of 857 reads per second), and only around 1,000 writes per second.

In contrast, Scylla still performed reasonably well under the same conditions: 20,200 operations per second with good 99% latency against a single partition.

TESTING AGAINST UNIFORM DATA DISTRIBUTION

Since DynamoDB is known to be tricky when data distribution isn't uniform, the tests were also run against a uniform distribution to test Dynamo within its 'sweet spot'. To demonstrate, the benchmark was run against 3 nodes of i3.8xl for Scylla, with replication of 3 and quorum consistency level, with the 1TB YCSB dataset (replicated 3 times).

In this test, DynamoDB met the throughput SLA of 120,000 operations per second. However, it failed to meet the latency SLA of 10ms for 99%.

Scylla, on the other hand, easily met the throughput SLA, with only 58% load and latency. That was 3x-4x better than DynamoDB and well below the requested SLA.

Scylla Enterprise Cluster	Amazon DynamoDB Provisioned Capacity
i3.8xlarge | 32 vCPU | 244 GiB | 4 x 1.9TB NVMe	160K write | 80K read (strong consistency)
3-node cluster on single DC | RF=3	Dataset: ~1.1TB (1B partitions / size: ~1.1Kb)
Dataset: ~1.1TB (1B partitions / size: ~1.1Kb)	Storage size: ~1.1 TB (DynamoDB table metrics)
Total used storage: ~3.3TB	

ADDITIONAL CONSIDERATIONS

CROSS-REGION REPLICATION AND GLOBAL TABLES

Globally distributed deployments are affected by replication speed between datacenters. A simple comparison showed that DynamoDB replicated in 370ms on average to a remote datacenter, while Scylla's averaged 82ms to accomplish the same task. Since DynamoDB's cross-region replication is built on its streaming API, it seems that congestion has the potential to grow into a multi-second gap.

Unlike DynamoDB, Scylla enables administrators to quickly add regions on demand with a single command:

```
ALTER KEYSPACE mykespace WITH replication
= { 'class' : 'NetworkTopologyStrategy',
'replication_factor': '3', '<exiting_dc>'
: 3, <new_dc> : 4};
```

In DynamoDB, on the other hand, global tables must be defined in advance. This imposes a significant obstacle to scalability, and a major cost as datacenters grow over time.

THE HIGH COST OF CACHING

DynamoDB provides an optional in-memory cache called DynamoDB Accelerator (DAX). DAX can improve performance, and DynamoDB's high cost can be reduced in some cases by using DAX. In contrast, Scylla has a much smarter and more efficient embedded cache (please read our earlier blog post for more details).

BEYOND THE LIMITS

DynamoDB imposes a 400KB size limit on the size of each cell. Scylla supports cells that can be measured in megabytes. In at least one production use case, Scylla has been deployed as a storage system for large blobs with single-digit millisecond latency.

DynamoDB cannot store items larger than 10GB, yet another problematic limit in DynamoDB caused by the way it designed its partition limits. While it's not a recommended pattern, some Scylla customers store 130GB items in a single partition. The effect of these higher limits is more freedom in data modeling and fewer limitations on future requirements.

ALTERNATOR: SCYLLA'S DYNAMODB-COMPATIBLE API

In 2019, Scylla introduced Project Alternator, open-source software that enables application- and API-level compatibility between Scylla and DynamoDB. With Alternator, DynamoDB users can seamlessly transition to Scylla Open Source for better performance, lower costs and no vendor lock-in.

Scylla's DynamoDB API can be deployed on premises, on public clouds like Microsoft Azure or Google Cloud Platform, or on AWS. On AWS Scylla users can take advantage of other aspects of the Amazon cloud ecosystem, such as high-density i3en instances. DynamoDB users can run existing client applications with no modifications. Alternator is written in C++ and is a part of Scylla.

Alternator gives developers greater control over large-scale, real-time big data deployments, starting with costs. A typical Scylla cluster will cost 10%-20% the expense of the equivalent DynamoDB table. Alternator also frees developers to access their data without limits by eliminating payment per operation — they can run as many operations as their clusters support, keeping costs low and predictable.

Alternator gives operators control over the number of replicas as well as the balance of cost versus redundancy to suit their applications. Operators can set and change the replica number per data center, the number of zones, and the consistency level on a per-query basis.

For more information on Project Alternator, please visit scylladb.com/alternator.

CONCLUSION

Selecting a NoSQL database can be a daunting task; the benefits must be clearly defined and the risks mitigated by whatever means necessary. The goal of this paper has been to provide a fair and factual basis for comparing Amazon's well-known DynamoDB against the relative newcomer, Scylla.

If you'd like to try your own comparison, remember that Scylla is open source. Feel free to download now. Please contact us if you have any questions about how we stack up or if you'd like to share your own results. Or start by running a cloud-hosted Scylla Test Drive, which lets you spin-up a hosted Scylla cluster in minutes. We'll end with a final reminder that our Scylla Cloud is built on Scylla Enterprise, delivering similar price-performance advantages while eliminating administrative overhead. A free trial of Scylla Cloud is available at cloud.scylladb.com/user/signup

ABOUT SCYLLADB

Scylla is the real-time big data database. A drop-in alternative to Apache Cassandra and Amazon DynamoDB, Scylla embraces a shared-nothing approach that increases throughput and storage capacity as much as 10X that of Cassandra. Comcast, Banco Santander, Samsung, Starbucks, Johnson & Johnson, Discord, Fanatics, FireEye, Lookout, Grab and many more leading companies have adopted Scylla to realize order-of-magnitude performance improvements and reduce hardware costs.

Scylla is available in Open Source, Enterprise and fully managed Cloud editions. ScyllaDB was founded by the team responsible for the KVM hypervisor and is backed by Bessemer Venture Partners, Eight Roads Ventures, Innovation Endeavors, Magma Venture Partners, Qualcomm Ventures, Samsung Ventures, TLV Partners, Western Digital Capital and Wing Venture Capital.

For more information: ScyllaDB.com

SCYLLADB.COM

SCYLLA

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