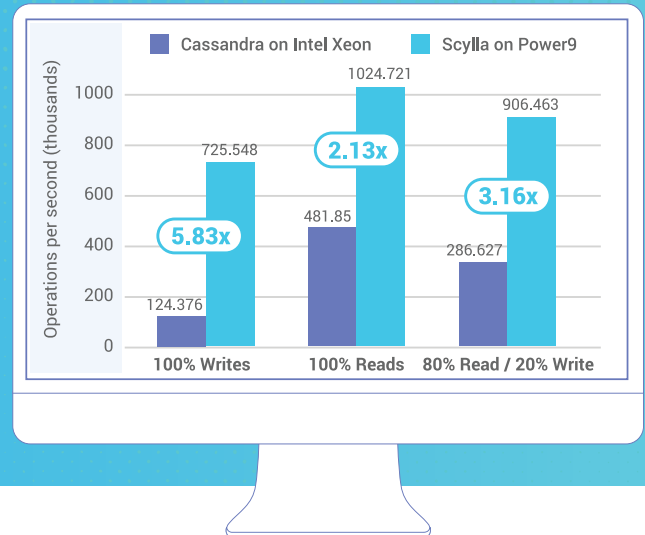


The Real-Time Big Data Database on IBM Power Systems

Scylla fully leverages the advantages of the IBM Power System multi-core architecture to deliver scale-up performance of 1,000,000 IOPS per node and 99% latency under 1 millisecond.



IBM benchmark shows throughput of Scylla running on IBM Power9 versus Cassandra running on Intel Xeon SP.

Scylla is trusted by internet and traditional enterprises like IBM, AppNexus, Ola Cabs, Investing.com, mParticle, and many others.



SAMSUNG SDS

musical.ly

adgear



THE BENEFITS OF SCYLLA



10x Higher Throughput. Written in C++ to maximize hardware utilization and achieve up to 1,000,000 read/write operations per node.



Always-On Availability. Automatic failover and replication across multiple nodes and data centers provide reliable fault tolerance.



Low and Consistent Latency. Lockless implementation and an independent memory management stack remove the inefficient reliance on JVM or Linux page cache and deliver consistently low latency.



Compaction, Streaming, and Repair Solved. Scylla's auto-tuning capabilities include an array of dynamic scheduling algorithms that minimize database operation latency jitter and reduce compaction streaming and repair time.



Highly Scalable. Auto-sharding, homogeneous servers, and native multi-datacenter implementation allow seamless linear scaling without compromising on application downtime or performance.



Community Backed. Scylla has been open sourced since day one and is backed by a growing community of contributors, and Scylla leverages the big data ecosystem around Cassandra, Spark, JanusGraph, etc.



Optimum Total Cost of Ownership. Scylla maximizes utilization of IBM Power—including CPU, memory, and disk—and network interfaces with linear scale-up out of the box.

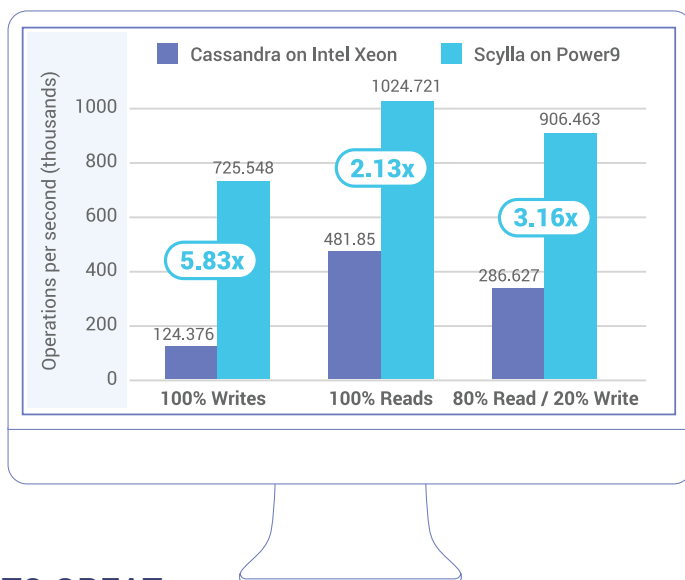


Easy to Use. Apache Cassandra's wire protocol, a rich polyglot of drivers, and integration with Spark, Presto, and Graph tools mean resource-efficient and performance-effective coding.

SCYLLA ON IBM POWER9 PERFORMANCE EXCEEDS CASSANDRA ON INTEL XEON SP

Recent performance benchmarks by IBM show...

- Scylla on IBM POWER9 achieves **nearly 6X the write throughput** of Cassandra on Intel Xeon SP
- Scylla on IBM POWER9 reaches **over 2X the read throughput** of Cassandra on Intel Xeon SP
- Scylla on IBM POWER9 delivers over **3X the throughput** of Cassandra on Intel Xeon SP for 80% reads, 20% writes.



TAKING BIG DATA DEPLOYMENTS FROM GOOD TO GREAT



Cassandra



ScyllaDB

High Availability	✓ Multi-region and fault tolerant	✓ Multi-region and fault tolerant
Scale Out	✓ Homogeneous nodes scale to hundreds of nodes per cluster	✓ Homogeneous nodes scale to hundreds of nodes per cluster. Additionally, Scylla scales up with the number of cores in our server
Scale Up	✗ Unpredictable and unbounded latency, mainly the result of the JVM's GC	✓ C++, shared-nothing and lockless novel design allows < 1 millisecond tail latency
Predictable Low Latency	✗ Limited per-node performance. Cannot fully exploit the disk, memory and CPUs	✓ Asynchronous core engine with shard-per-core architecture allows perfect scale-up
Low Management	✗ Intricate tuning, endless compactions and GC storms makes it hard to maintain	✓ Auto-tuning, isolation and prioritized workloads minimize operational overhead

GET STARTED TODAY

Spin up a live Scylla cluster and experience extreme performance first-hand at scylladb.com/test-drive.

For more information, please visit scylladb.com.

“We spent a lot of time on the other databases—mainly Hbase and Cassandra—to try to make them more comparable to Scylla. But of course we failed miserably. They still don’t perform as well as Scylla.”

Ted Chang and Chin Huang, *IBM Solution Architects*

SCYLLA.

United States Headquarters
1900 Embarcadero Rd
Palo Alto, CA 94303 U.S.A.
Phone: +1 (650) 332-9582
Email: info@scylladb.com

Israel Headquarters
11 Galgalei Haplada
Herzeliya, Israel



Copyright © 2018 ScyllaDB Inc. All rights reserved. All trademarks or registered trademarks used herein are property of their respective owners.