
Rethinking the database with drop-in replacements

Abhishek Tiwari 

Citation: *A. Tiwari*, "Rethinking the database with drop-in replacements",
Abhishek Tiwari, 2015. [doi:10.59350/sy6ep-jcf09](https://doi.org/10.59350/sy6ep-jcf09)

Published on: September 27, 2015

We are observing a new era in database technology space, drop-in replacements with order of magnitude improvement in performance over their counterparts. First Amazon Aurora and now Scylla DB. Although the concept of drop-in replacements is not something new but most recently these substitutes started pushing the performance boundaries using completely overhauled architecture. Apart from feature parity, the focus for drop-in replacements is shifted towards lowering operational and infrastructure costs with the ability to scale and heal.

Amazon Aurora

As [discussed before](#) on [this blog](#), Amazon Aurora is the fully compatible substitute for MySQL 5.6 with 5x increase in throughput performance over stock MySQL 5.6 on the similar hardware. In addition, Amazon Aurora reduces latency often down to single digit milliseconds. Aurora's fault-tolerance and self-healing mechanics can be considered as best in class. Before Amazon Aurora, MySQL had at least two popular enhanced drop-in substitutes - MariaDB and Percona. But as [elaborated here](#) Amazon Aurora is a game changer.

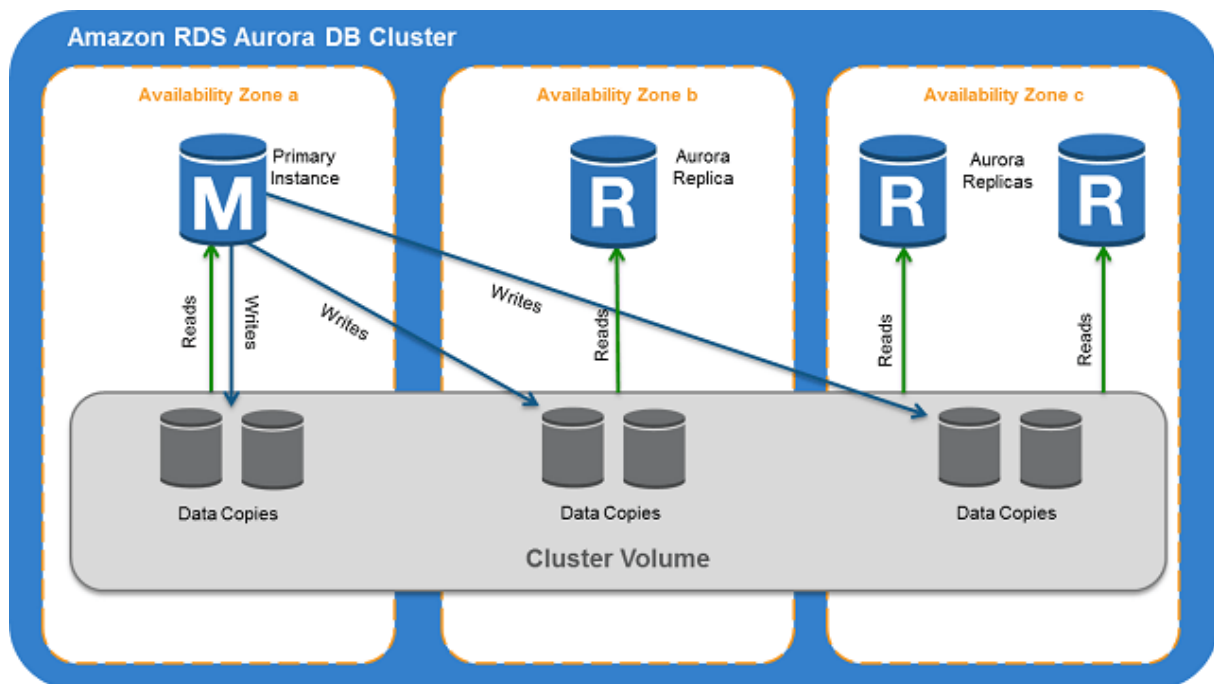


Figure 1: Amazon Aurora Architecture. Credits AWS

Scylla drop-in replacement for Cassandra

Scylla which claims to be the world's fastest column-store database is a drop-in replacement for popular NoSQL database technology Apache Cassandra 2.1.9. Scylla supports Cassandra data format (SSTable), configuration, and all relevant external interfaces. Key Cassandra tools (CQLSh, cassandra-stress, nodetool) are either fully or partially compatible. Scylla supports the majority of CQL keyspace, and table operations, but currently it lacks support for ALTER TABLE, DROP TABLE, TRUNCATE and pagination. Similarly, support for the multiple data centers, which is a key Cassandra feature, is a work in progress.

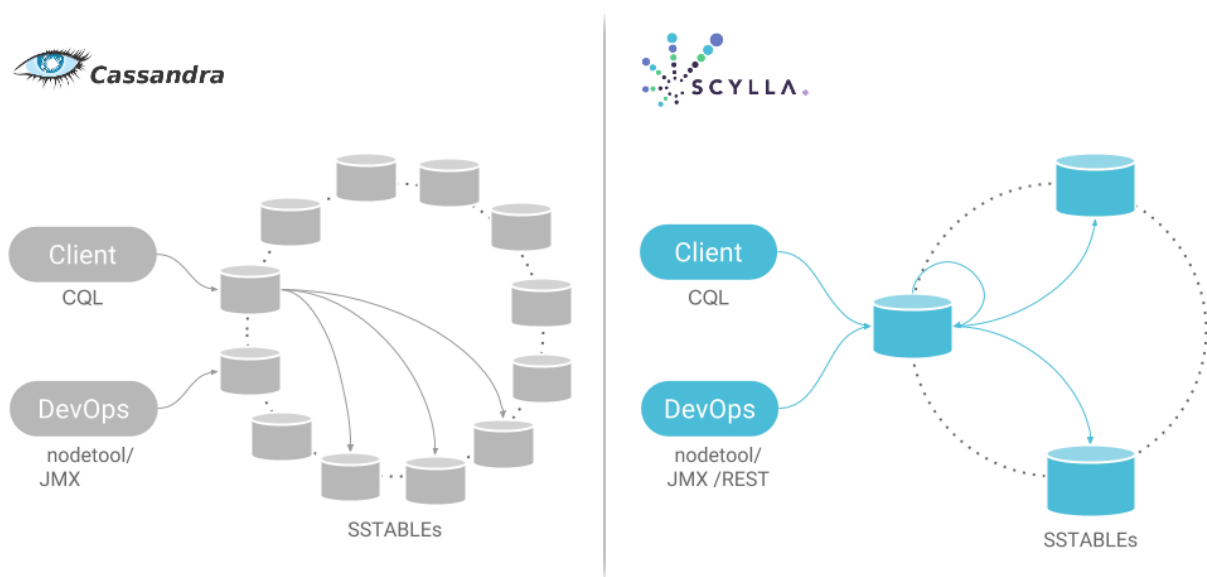


Figure 2: Scylla as drop-in replacement solution for Cassandra. Credits ScyllaDB.com

Scylla architecture

Scylla architecture is based on a shared-nothing approach at individual core level which means Scylla avoids the core locking, and latency due coordination across cores. To achieve the extreme performance on multicore hardware, Scylla uses the Seastar framework. Basically, Scylla runs multiple engines, one per core, each with its own memory, CPU, and multi-queue NIC which allows to squeeze the most out of the hardware. For those who are unfamiliar with Seastar, it is an advanced, open-source C++ framework for high-performance server applications on modern hardware.

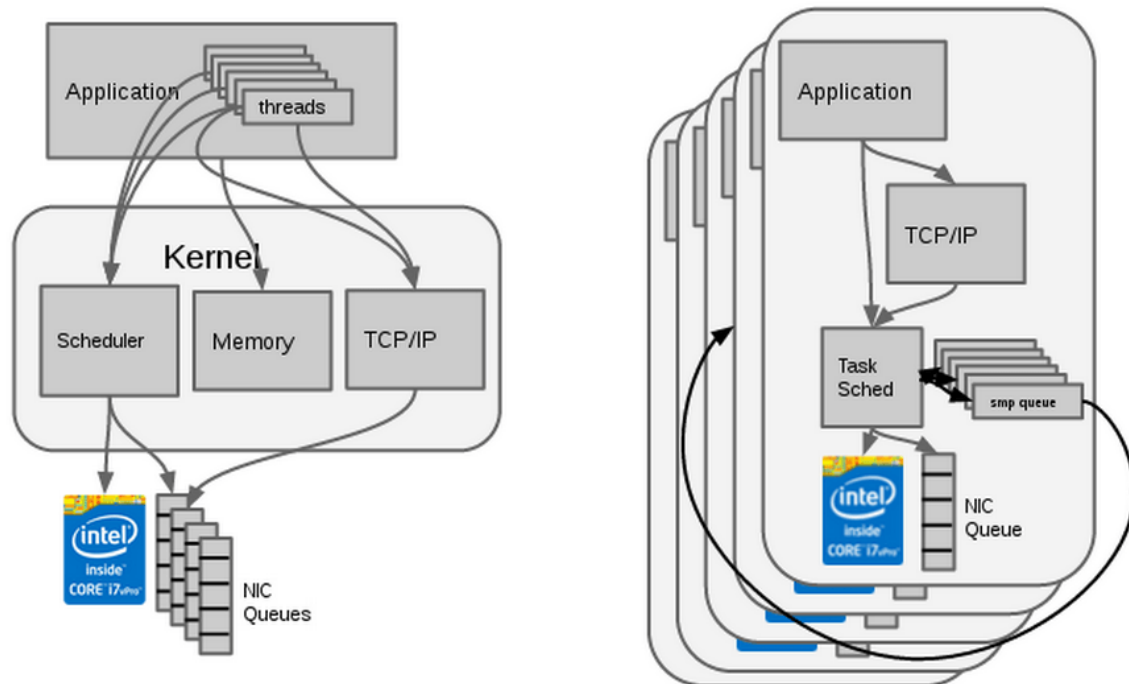


Figure 3: The Seastar model, shared-nothing cores and lockless inter-core communication. Credits ScyllaDB.com

Scylla benchmarks

The following benchmark compares Cassandra and Scylla throughput on a single server. Benchmark suggests Scylla can achieve up to 1 million transactions/second per server. In addition, Scylla's measured consistent latency under 1ms (99th percentile) significantly lower than Cassandra's, while providing significantly higher throughput.

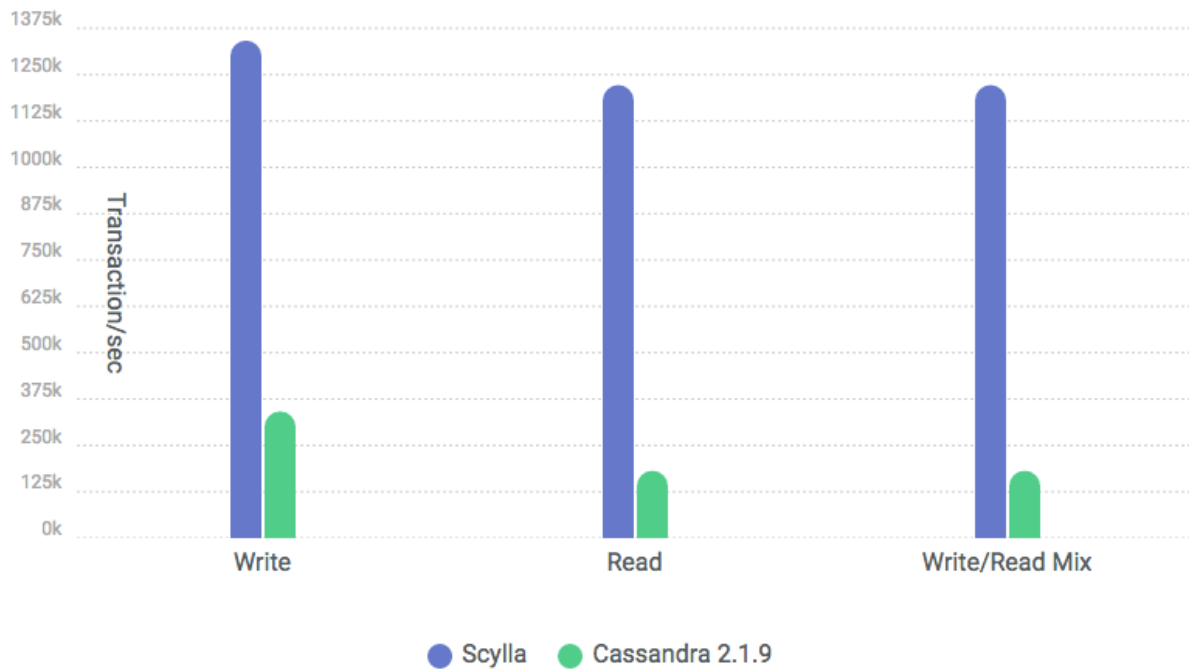


Figure 4: Throughput of Scylla and Cassandra on a single multi-core server. Average throughput. Results were rounded with the accuracy of 10K. Credits ScyllaDB.com

Closing thoughts

These innovations are not just driving the performance to next level but also cutting the total cost to achieve the required throughput and latency. You pay much less towards the infrastructure cost despite the order of magnitude improvement in performance and latency.