Scylla: A new NoSQL Database

Capable of 1,000,000 operations per second
PER NODE
With predictable, low latencies
Compatible with Apache Cassandra drivers, integration, management tools
THROUGHPUT

Scylla Performance

Transactions per second

Write

Read

Mixed

Load

Scylla 0.9

C* 2.1.9
Scylla and Cassandra Latency

Latency in MS

- Latency mean
- Latency median
- Latency 95th percentile
- Latency 99th percentile
- Latency 99.9th percentile

Scylla Latency
Cassandra Latency
Scylla Throughput
Cassandra Throughput

throughput
Average throughput. Results were rounded with accuracy of 10K.
FULLY COMPATIBLE

- Uses Cassandra SSTables
- Use your existing drivers
- Use your existing CQL queries
- Use your existing `cassandra.yaml`
- Manage with `nodetool` or other JMX console
- Use your existing code with no change
- Copy over a complete Cassandra database
- Works with the Cassandra ecosystem (Spark etc.)
FULLY COMPATIBLE
SCYLLA IS QUITE DIFFERENT

Shard-per-core, no locks, no threads, zero-copy
Based on the Seastar C++ application framework
Efficient, unified DB cache (vs. Linux page cache)
CQL-oriented storage engine
Exploit all hardware resources - NUMA, multiqueue NICs, etc
KVM was invented by Avi in 2006, development was managed by Dor. It was a new hypervisor after VMW, Xen had dominated the market. By smart design choices and leveraging Linux and the hardware, it became the most performing hypervisor. KVM holds SPECvirt performance record and max IOPS record.

The Open Virtualization Alliance includes hundreds of companies, including HP, IBM, Intel, AMD, Red Hat, etc. KVM is the engine behind many clouds such as OpenStack, IBM, NTT, Fujitsu, HP, Google, DigitalOcean, etc.

### Traditional stack

- **Cassandra**
- **Scheduler**
- **Memory**
- **TCP/IP**

### Scylla sharded stack

- **Core Database**
- **Task Scheduler**
- **DPDK**

- **Kernel** (isn't involved)

No contention

- **Linear scaling**
- **NUMA friendly**

- **Lock contention**
- **Cache contention**
- **NUMA unfriendly**

- **Userspace**
- **NIC Queue**
- **SMP queue**
Scylla Memory Management
Seastar framework

- Each engine is executed by each core
- Shared-nothing per-core design
- No threads, no context switch and no locks
  - instead: asynchronous lambda invocation
- Programming model
  - Futures
  - Promises
  - Continuations
- Full kernel bypass, supports zero-copy
OSv: a library operating system

Unikernel

- C++ kernel code
- ZFS
- TCP/IP
- ifconfig.so
- Scylla.so

virtual memory

dynamic linker

“kernel threads”

“user threads”

single address space in “kernel mode”

disk

ifconfig.so

TCP/IP

ZFS

iface

virtio
Open source

URLs to remember:
- http://www.scylladb.com/
- http://seastar-project.com
- http://osv.io/