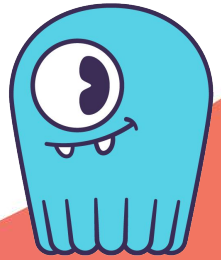
PERCONA LIVEONLINE

MAY 12 - 13th 2021

Shards all the way down

Building fast and highly concurrent databases on modern hardware



~$ whoami

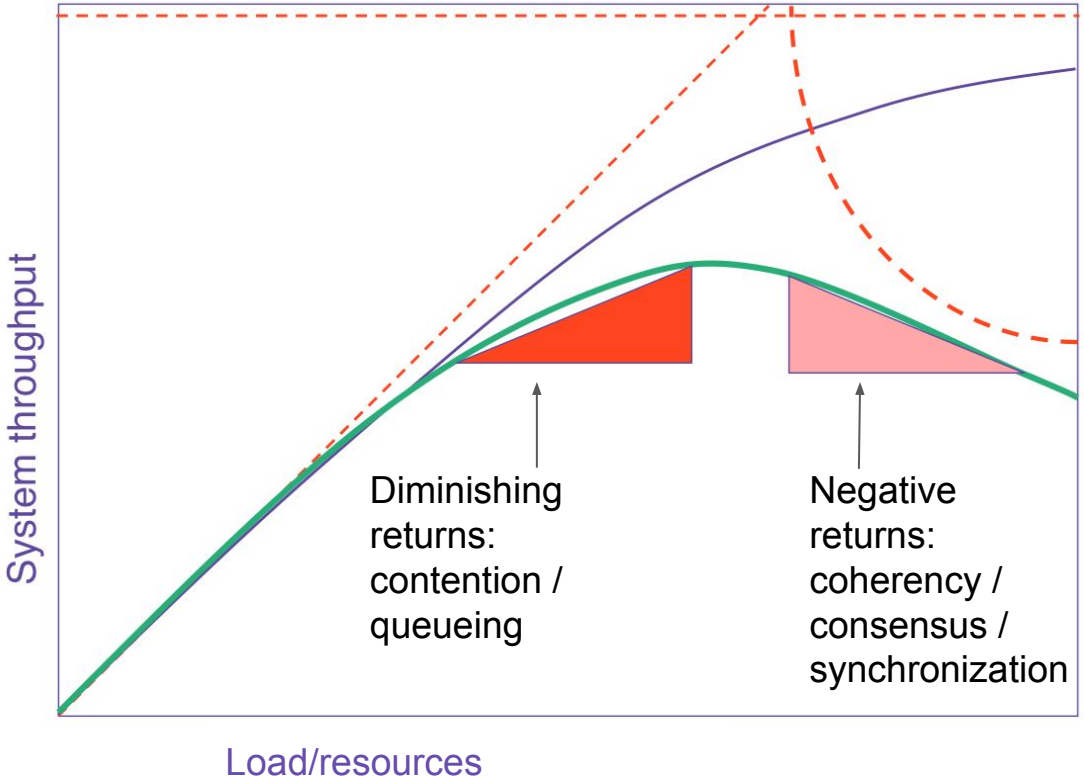
Avishai Ish-Shalom (@nukemberg)

Developer Advocate @ ScyllaDB

About ScyllaDB

- The Real-Time Big Data Database
- Drop-in replacement for Apache Cassandra and Amazon DynamoDB
- 10X the performance & low tail latency
- Open Source, Enterprise and Cloud options
- Founded by the creators of KVM hypervisor
- HQs: Palo Alto, CA, USA; Herzelia, Israel; Warsaw, Poland

The universal scalability law (USL)

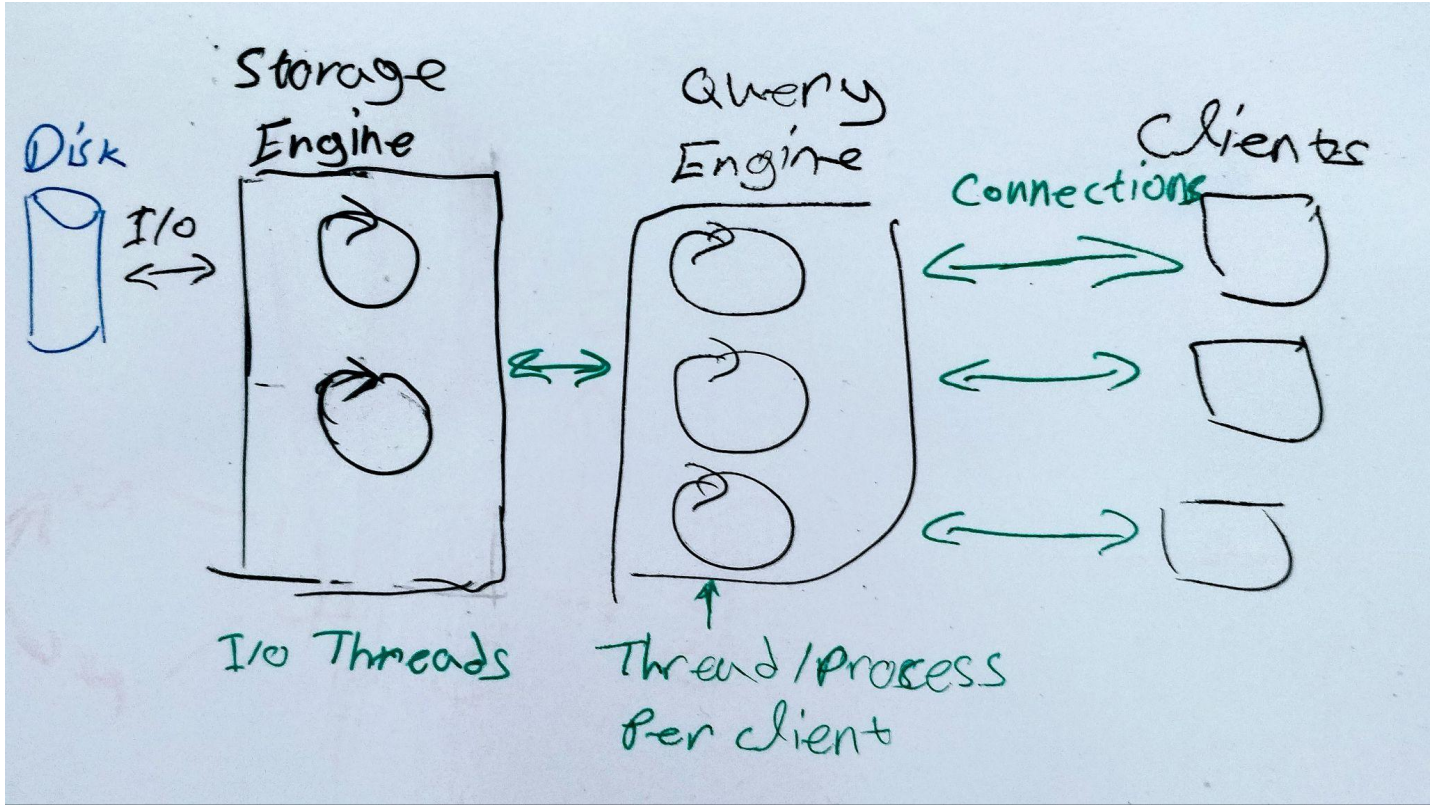


1.

How a database is built

Or at least, MySQL/Postgres et-al

Basic architecture

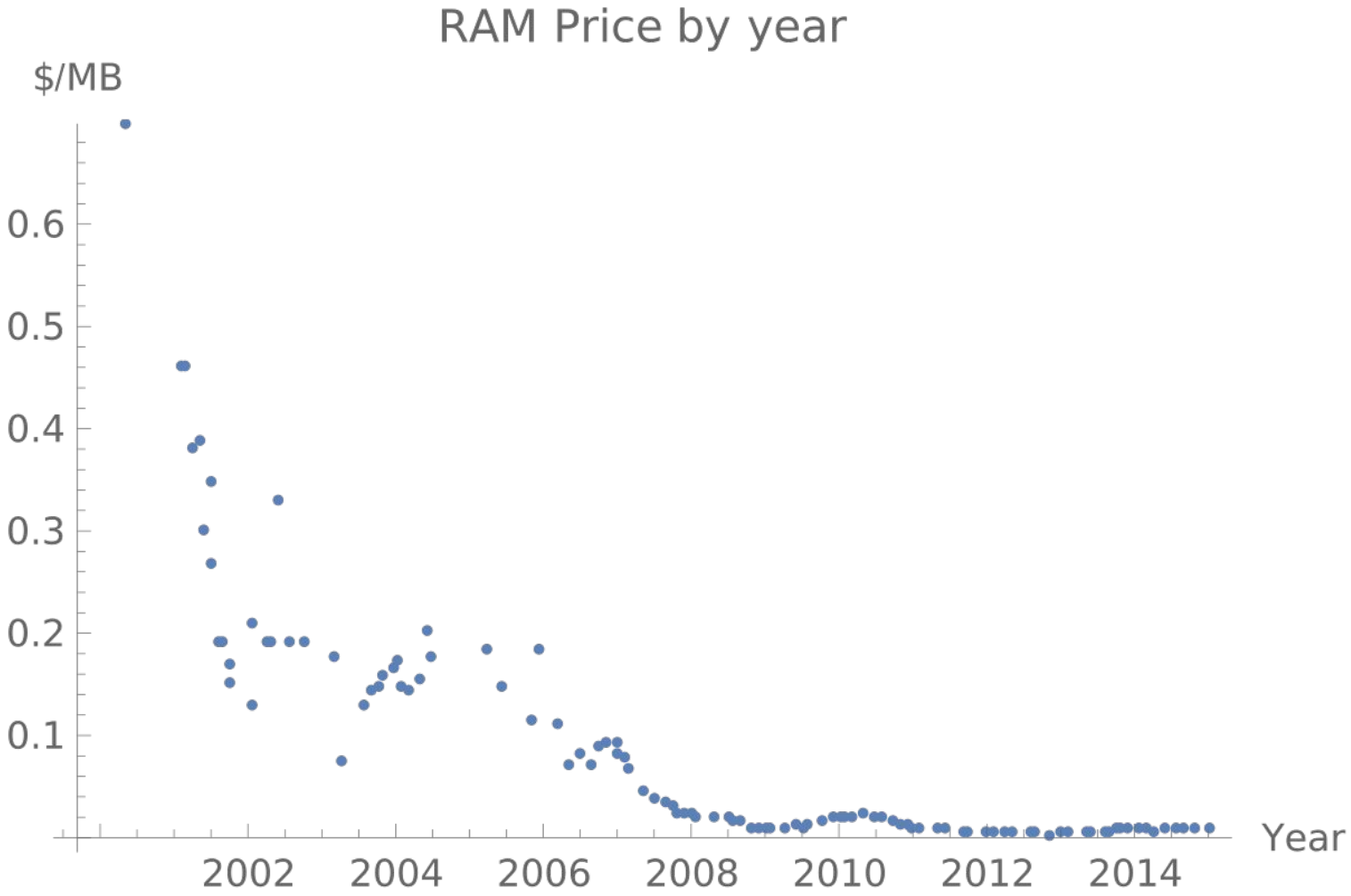


Basic architecture

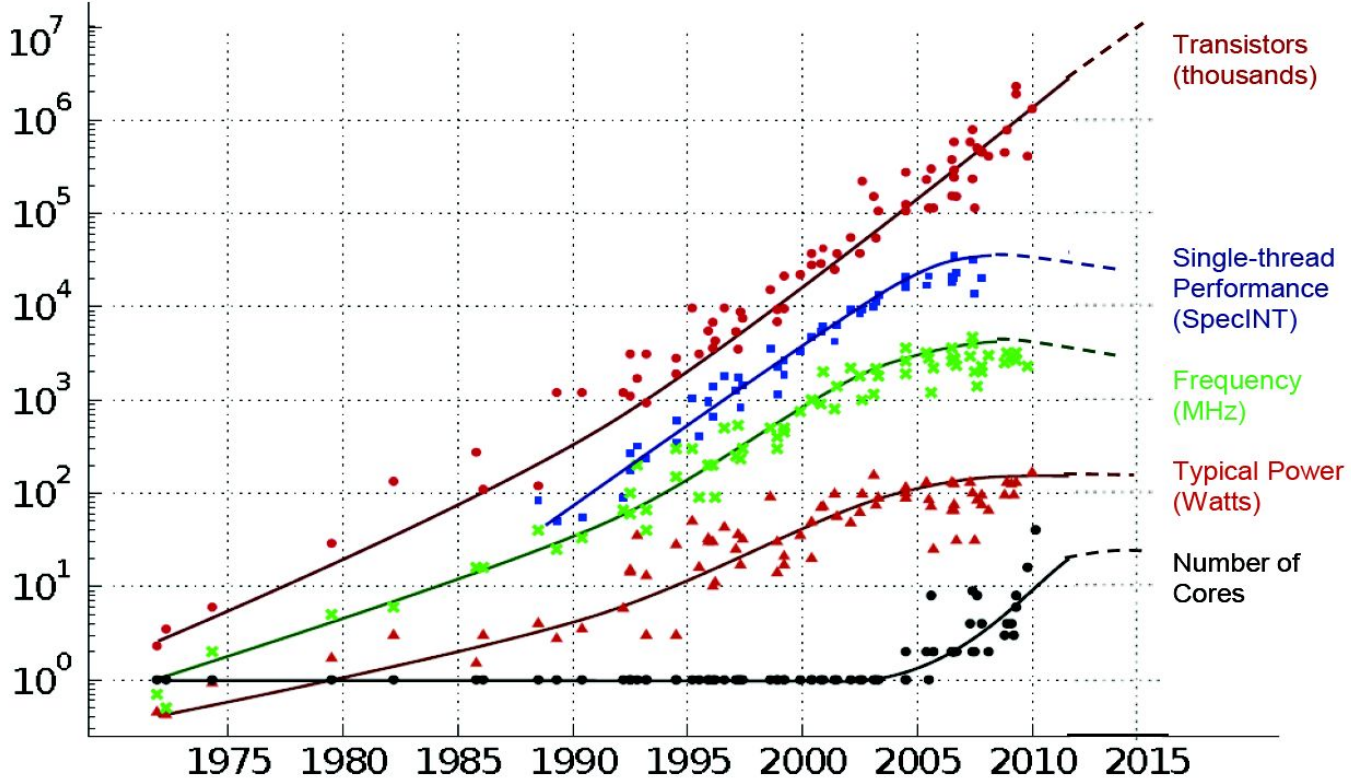
- Process/thread per client connection
- N storage/IO threads
- Thread MUTEX Locks to maintain storage consistency

Standard “Shared memory” architecture

2.
20 years of hardware evolution in 5 minutes


RAM Price by year
$/MB
0.6
0.5
0.4
0.3
0.2
0.1
2002
2004
2006
2008
2010
2012
2014
Year

35 YEARS OF MICROPROCESSOR TREND DATA



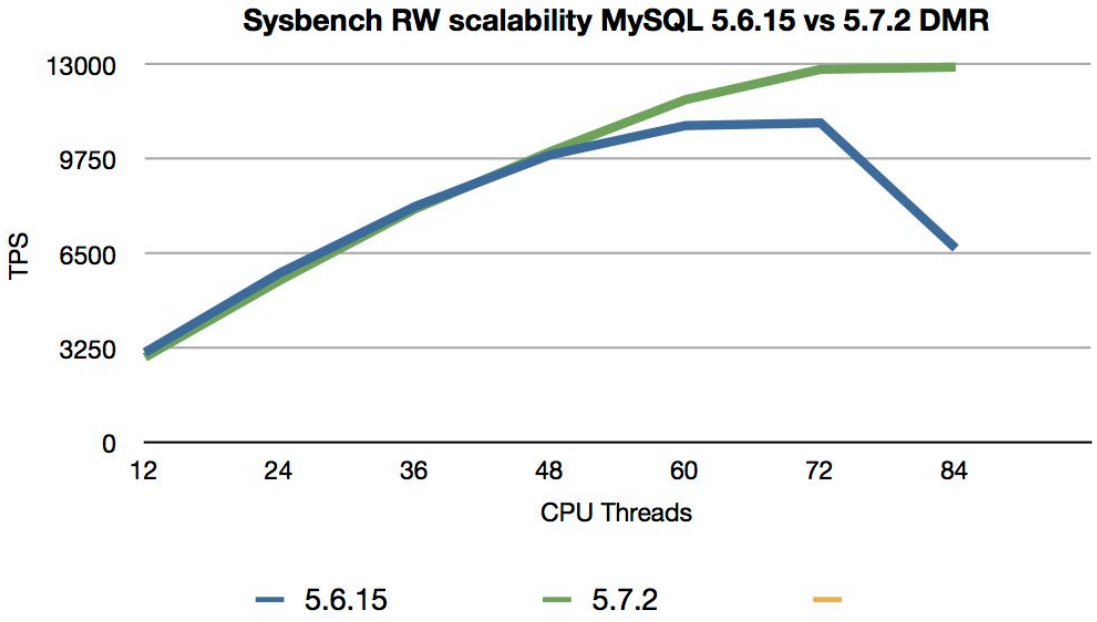
Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten
Dotted line extrapolations by C. Moore

What happened?

- Per thread performance plateaued
- Cores: 1 => 256
- RAM: 2GB => 2TB
- Disk space: 10GB => 10TB
- Disk seek time: 10-20ms => 20μs
- Network throughput: 1Gbps => 100Gbps

AWS u-24tb1.metal: 224 cores, 448 threads, 24TB RAM

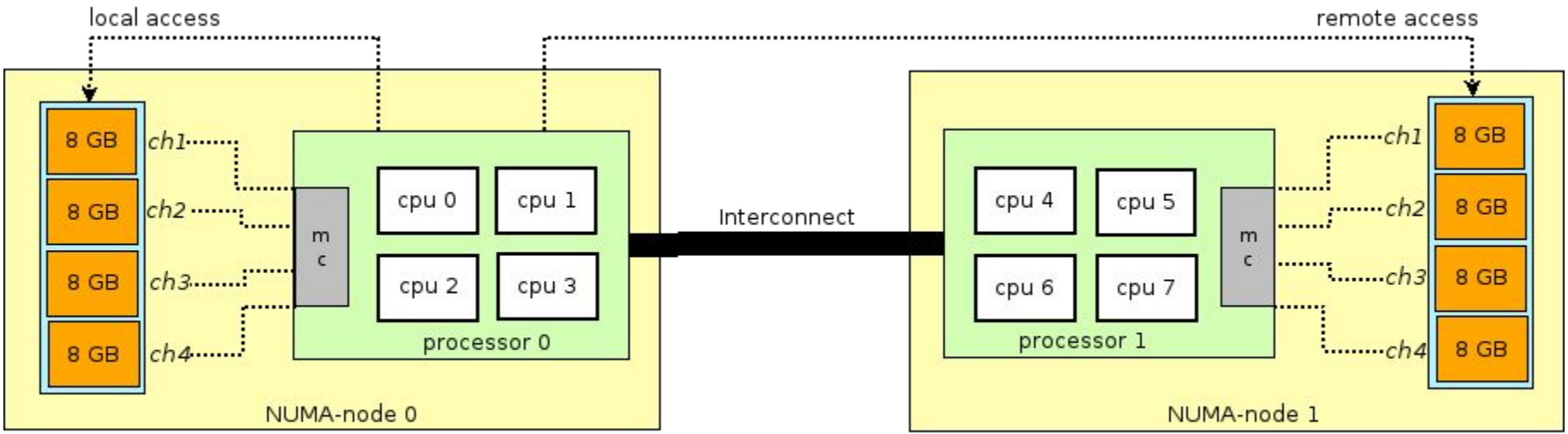
Remember the USL?



What’s going on?

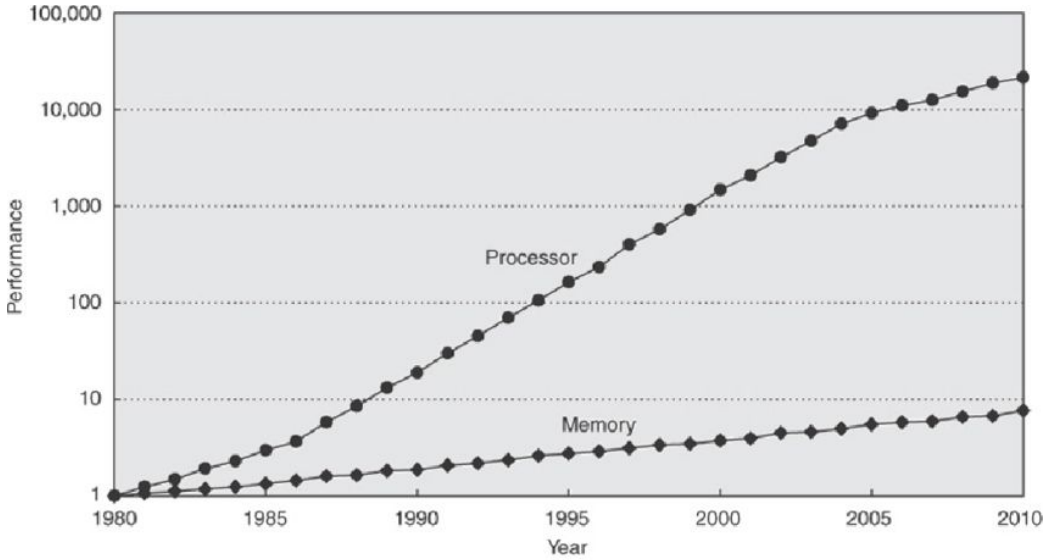
- MySQL max out around 48 cores
- Context switch ~1-2μs
- 10 Context switches is a missed disk seek
- Locks, locks and damn locks
- Because shared memory

Non Uniform Memory Access (NUMA)



The CPU-RAM-storage gap

- Memory seek is ~100 CPU cycles
- NVMe seek is ~1000 memory seeks



3.

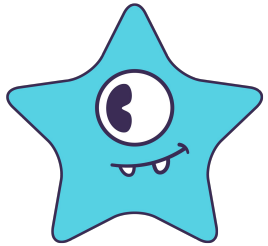
The database, reimagined

Let’s start from first principles

How do we use the hardware?

- No locks
- No shared memory
- No coordination/synchronization
- No context switches
- No memory copies

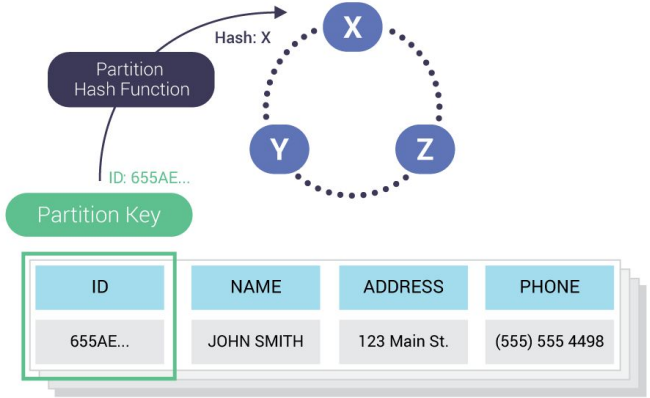
Shard per core



Share nothing, block nothing

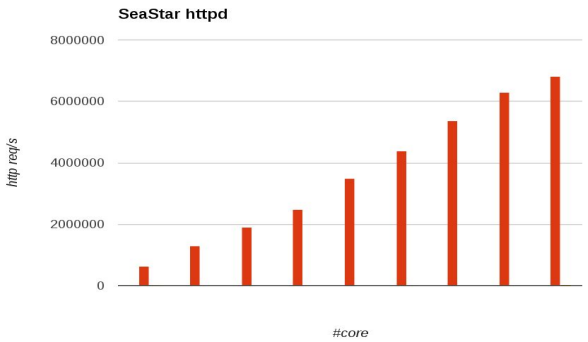
Sharding/partitioning

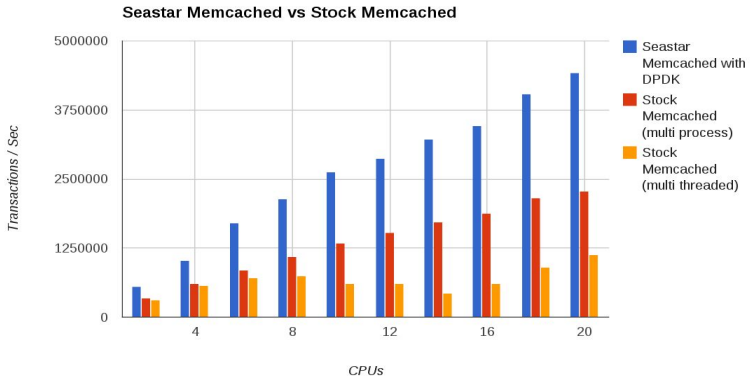
- Common concept in distributed databases
- Break the system to N non-interacting parts
- Usually done by `hash(partition_key) % N`
- Data/load may be unbalanced
  - Fact of life in distributed databases 🤷
  - Logical mapping of data shards to core shards



Seastar

- Open source framework, powering Scylla, RedPanda, ValueStore
- A “mini operating system in userspace”
- Task scheduler, I/O scheduler
- Fully asynchronous - userspace coroutines
- Direct I/O, self managed cache (bypass pagecache)
- One thread per core, one shard per core

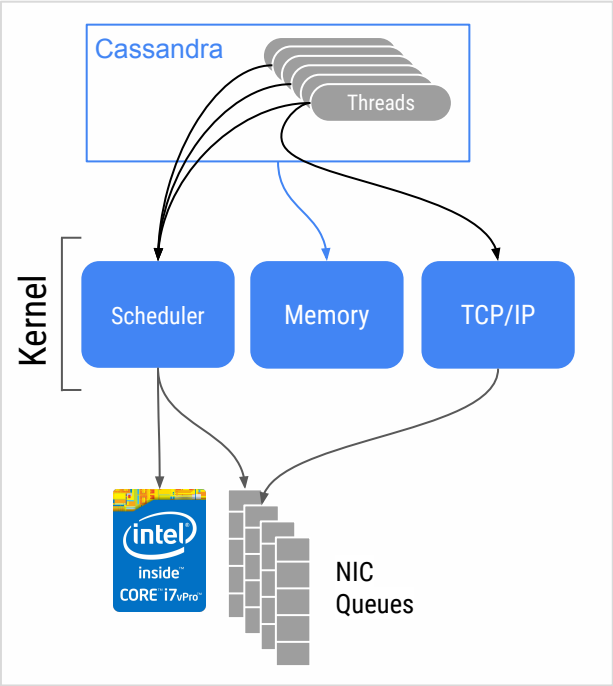




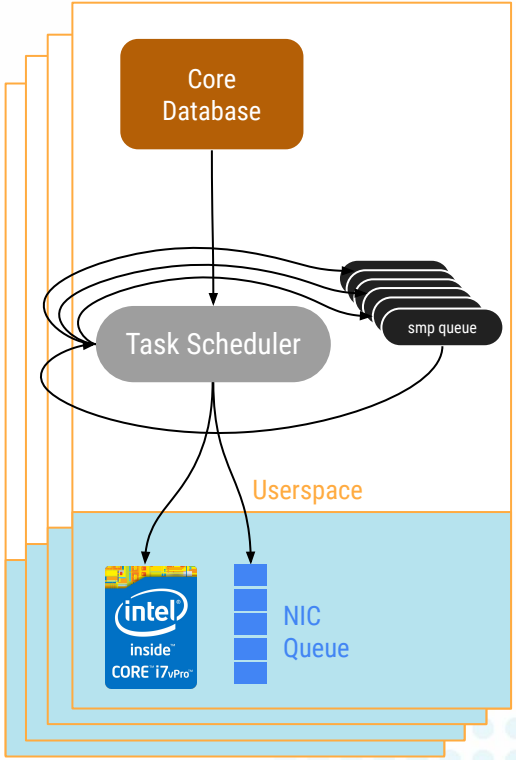
Shard per Core

Lock contention
Cache contention
NUMA unfriendly

Traditional Stack



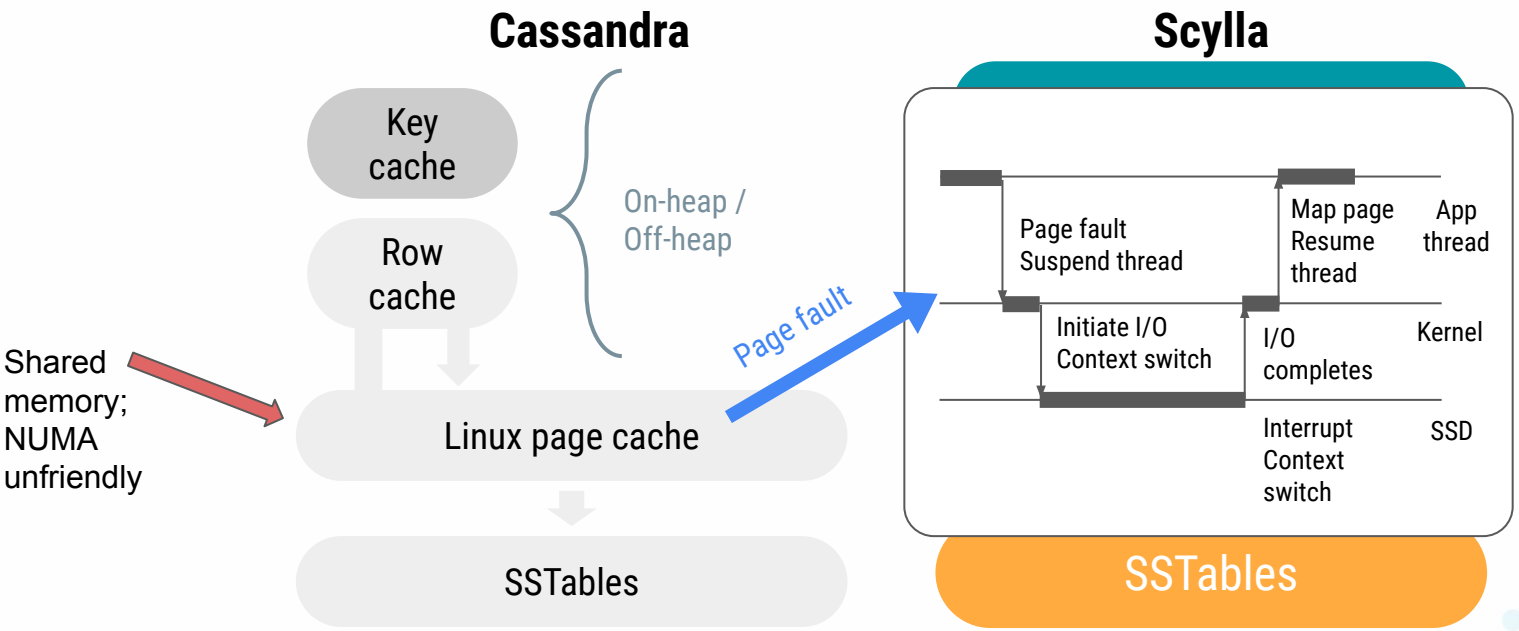
SeaStar's Sharded Stack



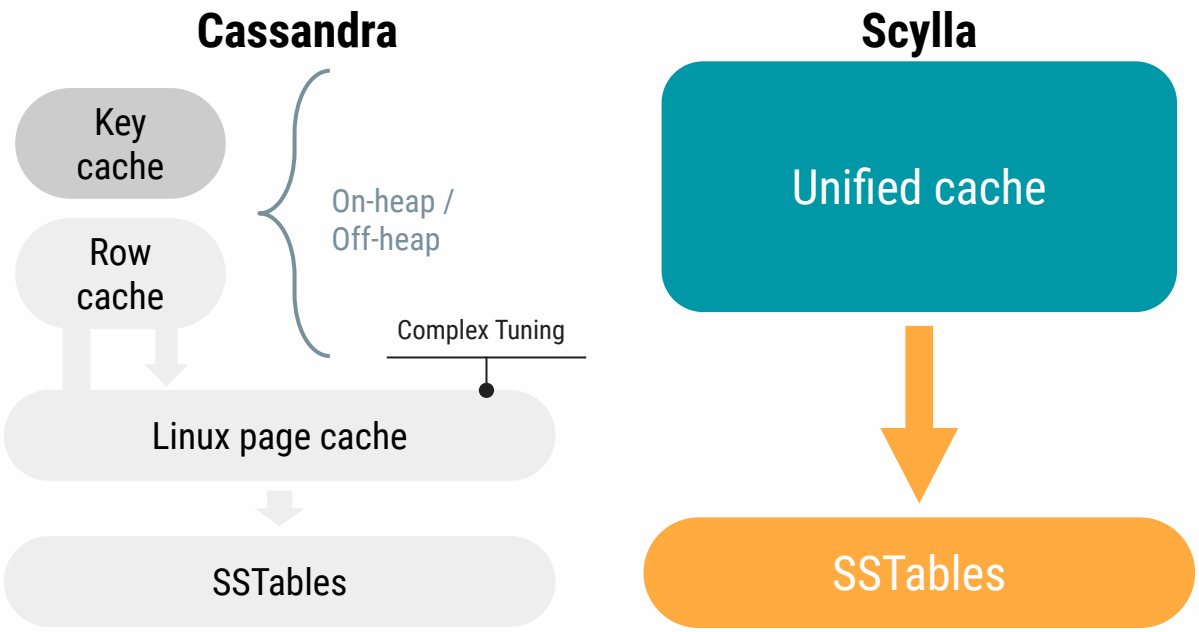
No contention
Linear scaling
NUMA friendly

SCYLLA.

Unified Cache



Unified Cache



Conclusion

Hardware

Changed

Software

Is the new bottleneck

Distributed

Architectures for the rescue

Thanks!

Any questions?

You can find me at:

- @nukemberg
- nukemberg@scylladb.com

THANK YOU !

PERCONA
LIVEONLINE
MAY 12 - 13th
2021