AEROSPIKE

A Real-time Operational Database: Aerospike

Brian Bulkowski
CTO, Founder
@bbulkow

Northern California Oracle Users Group
November 17, 2016

What is Aerospike ?

Large-scale DHT Database (10B ++ objects, 100T++, **O(1) get / put**)
… with queries, data structures, UDF, fast clients ...
... On Linux ...

High availability **clustering & rebalancing** (proven 5 9's, no load balancer)

Very **high performance C code** – reads and writes
(2M++ TPS from Flash, 4M++ TPS from DRAM ***PER SERVER***)

KVS++ provides **query, UDF, table/columns, aggregations, SQL**

Direct attach storage; persistence through replication and **Flash**

Cloud-savvy – runs with **EC2, GCE others; Docker**, more …

Dual License: Open Source for devs, Enterprise for deployment

AEROSPIKE

Enterprise Requirement: 2-Speed IT

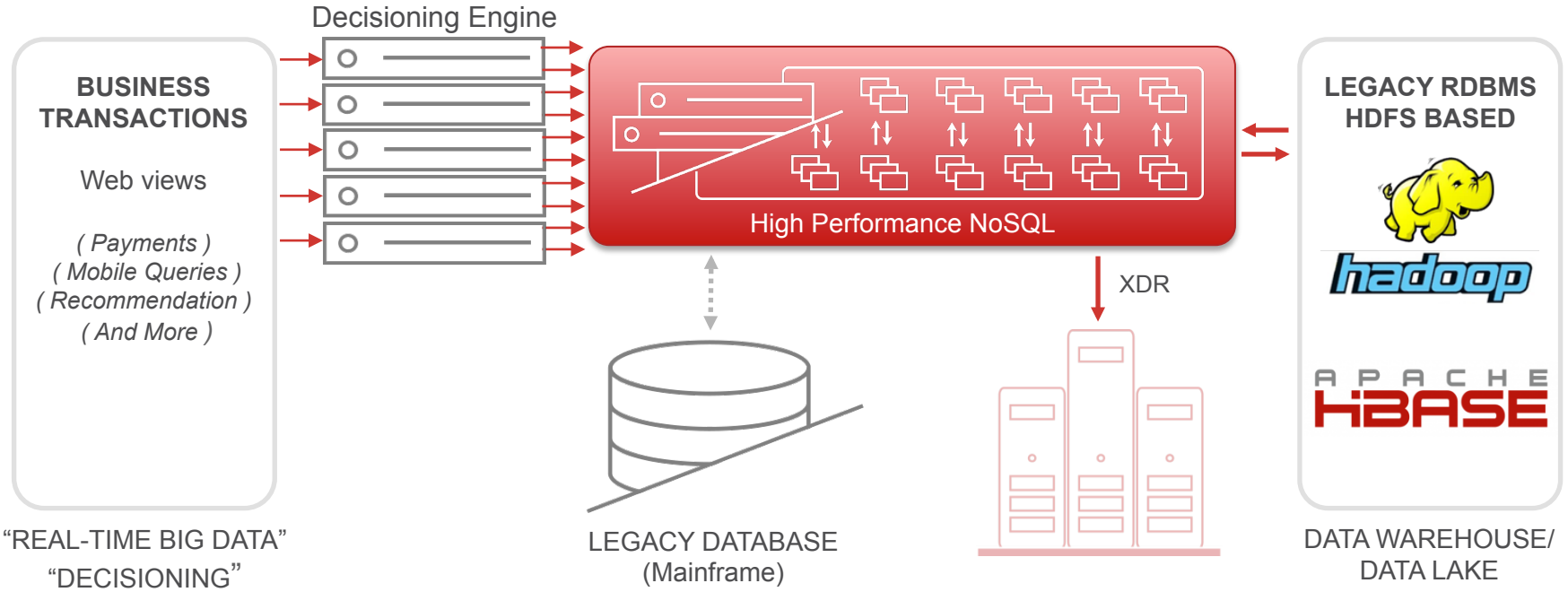
The only way for traditional enterprises to easily build Digital business is adapting to 2-Speed IT decoupling Systems of Record and Systems of Engagement

Front office Consumer Scale Digital Applications that move at a Faster pace and act as Systems of Engagement

Back office legacy Enterprise Scale Applications that move at a slower pace and act as Systems of Record

AEROSPIKE

Architecture Overview – Flash based system of engagement



500	X	5000	=	2.5 M
Business Trans per sec		Calculations per sec		Database Transactions per sec

AEROSPIKE

Real-world Engagements

AEROSPIKE

AdTech – Real-Time Bidding

Challenge

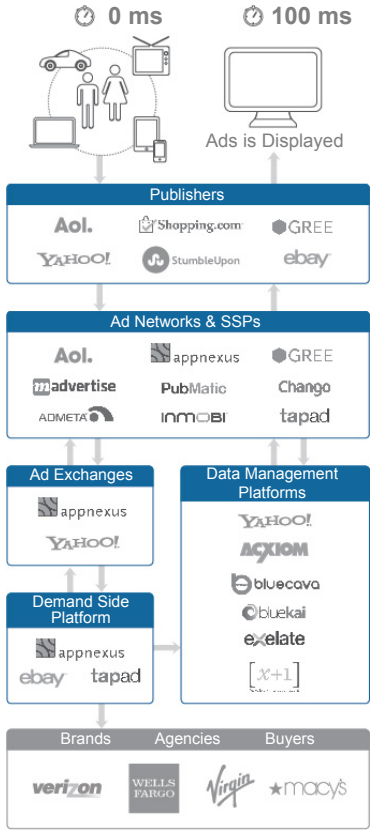
- Low read latency (milliseconds)
- 100K to 5M operations/second
- Ensure 100% uptime
- Provide global data replication

Performance achieved

- 1 to 6 billion cookies tracked
- 5.0M auctions per second
- 100ms ad rendering, 50ms real-time bidding, 1ms database access
- 1.5KB median object size

Selected NoSQL

- 10X fewer nodes
- 10X better TCO
- 20X better read latency
- High throughput at low latency



Fraud Prevention

Challenge

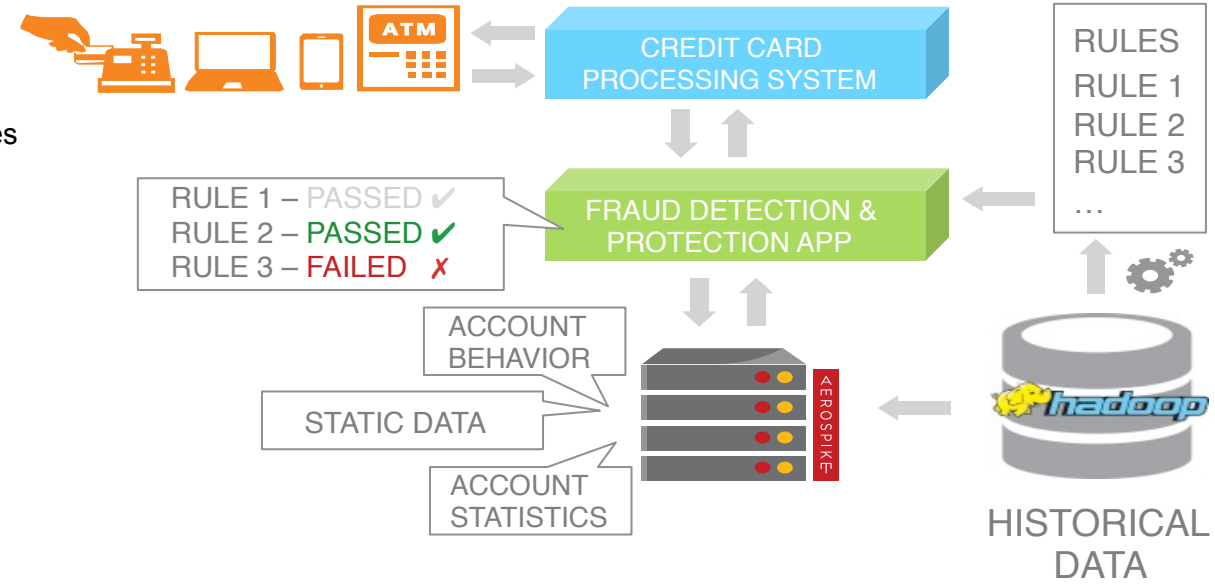
- Overall SLA 750 ms
- Loss of Business due to latency
- Every Credit Card transaction requires hundreds of DB reads/writes

Need to scale reliably

- 10 → 100 TB
- 10B → 100 B objects
- 200k → I Million+ TPS

Selected NoSQL

- Built for Flash
- Predictable Low latency at High Throughput
- Immediate consistency, no data loss
- Cross data center (XDR) support
- 20 Server Cluster
- Dell 730xd w/ 4NVMe SSDs



Fin Serv – Positions System of Record

Challenge

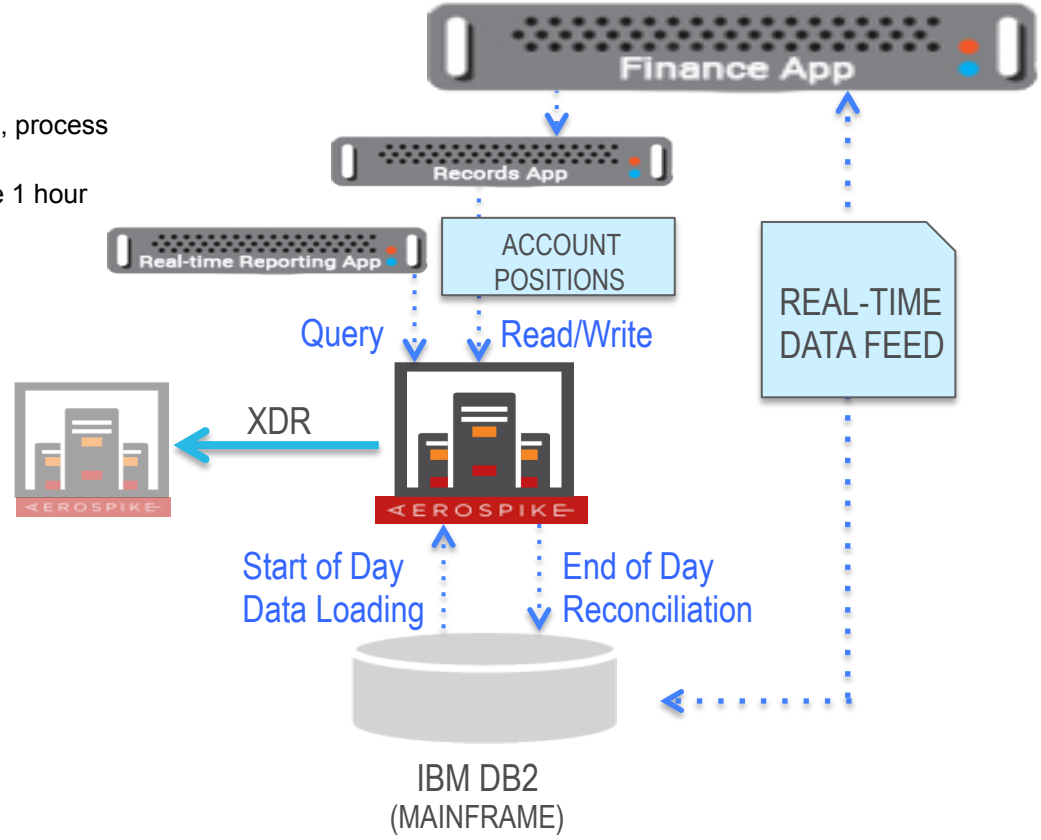
- DB2 stores positions for 10 Million customers
- Must update stock prices, show balances on 300 positions, process 250M transactions, 2 M updates/day
- Running out of memory, data inconsistencies, restarts take 1 hour
- 150 Servers -> Growing to 1000

Need to scale reliably

- 3 → 13 TB
- 100 → 400 Million objects
- 200k → I Million TPS

Selected NoSQL

- Built for Flash
- Predictable Low latency at High Throughput
- Immediate consistency, , no data loss
- Cross data center (XDR) support
- 10 Server Cluster



Telco – Real-Time Billing and Charging Systems

Challenge

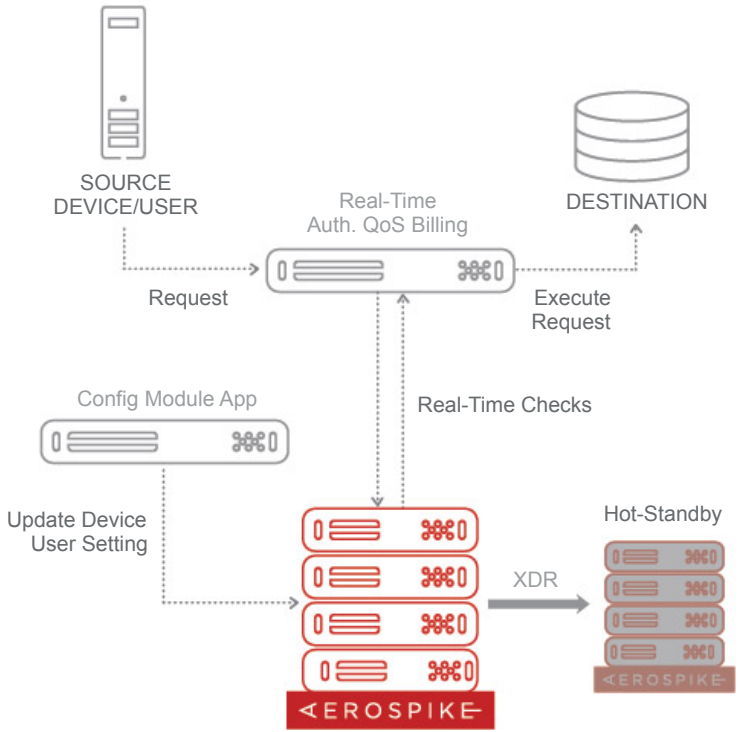
- Edge access to regulate traffic
- Accessible using provisioning applications (self-serve and through support personnel)

Need for Extremely High Availability, Reliably, Low latency

- > TBs of data
- 10-100M objects
- 10-200K TPS

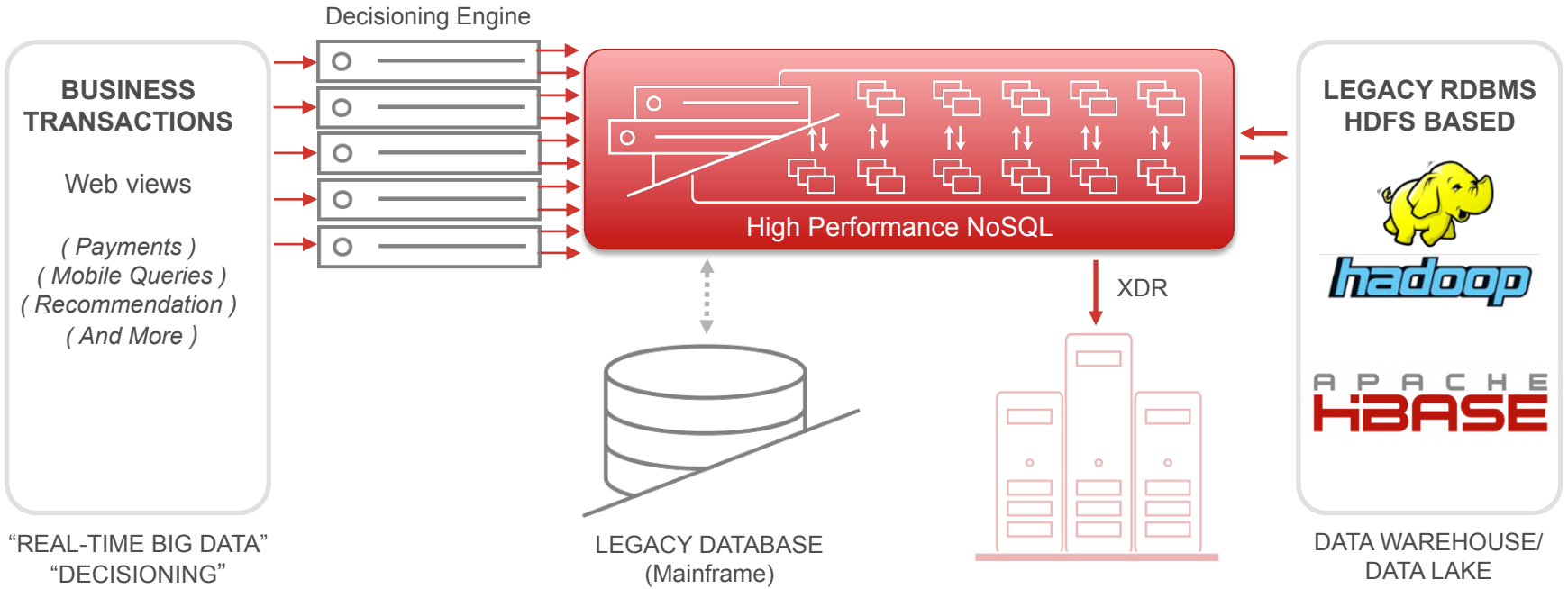
Selected NoSQL

- Clustered system
- Predictable low latency at high throughput
- Highly-available and reliable on failure
- Cross data center (XDR) support



AEROSPIKE

Operational Scale



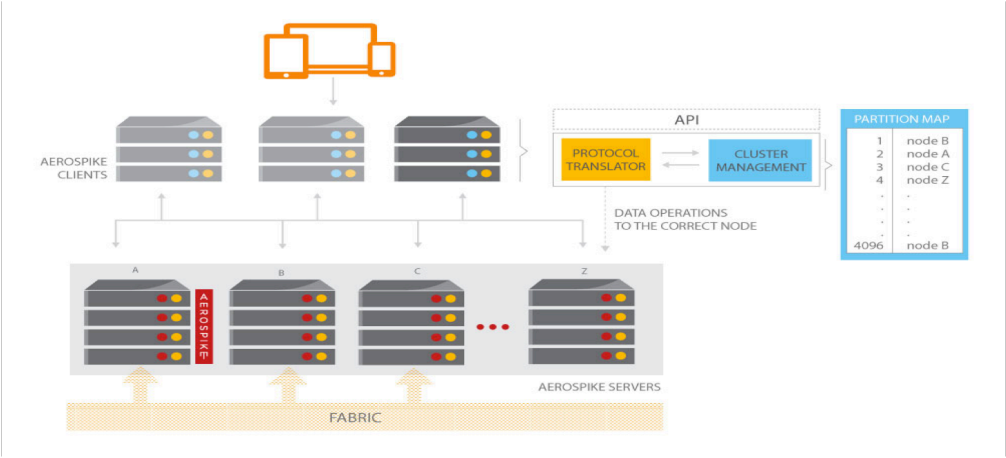
500	X	5000	=	2.5 M
Business Trans per sec		Calculations per sec		Database Transactions per sec

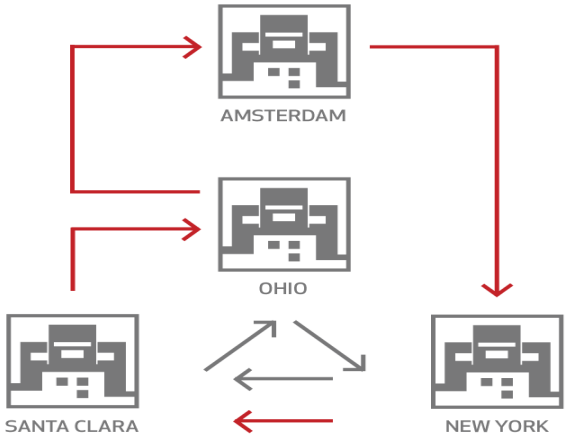
AEROSPIKE

Technology

AEROSPIKE

Architecture Overview





1) **No Hotspots**
   – Distributed Hashing simplifies data partitioning

2) Smart Client – **1 hop** to data, load balancing

3) **Shared Nothing Architecture**
   – every node is identical

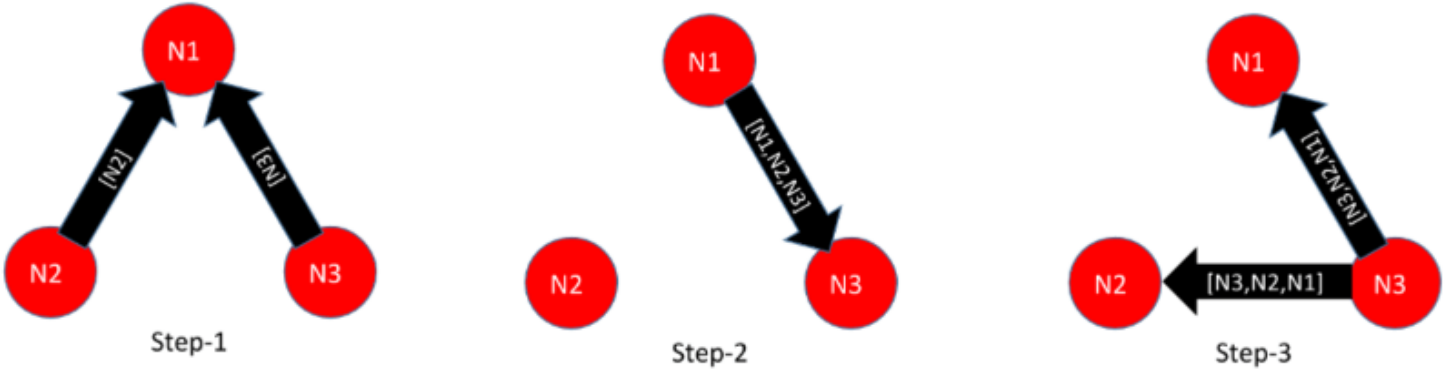
4) Smart Clustering
   – auto-sharding, auto-failover, auto-rebalancing, rack aware, rolling upgrades

5) Transactions and long-running tasks **prioritized in real-time**

6) XDR – sync replication across data centers ensures –near **Zero Downtime**

AEROSPIKE

Cluster Formation

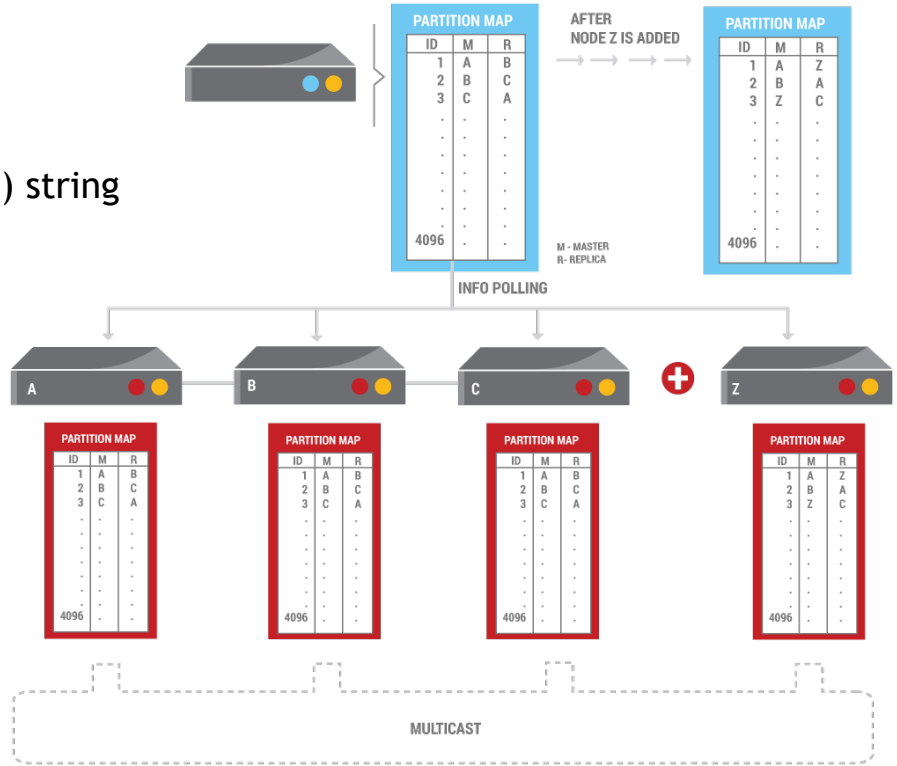


- Say N1 is seed node and N3 is the paxos principal
- N2 and N3 send themselves in list to N1; N1 discovers them
- N1 sends adjacency list [N1, N2, N3] to newly discovered node N3 (and also N2)
- N3 discovers N2 and starts sending the cluster node list [N3, N2, N1] to N1 and N2

AEROSPIKE

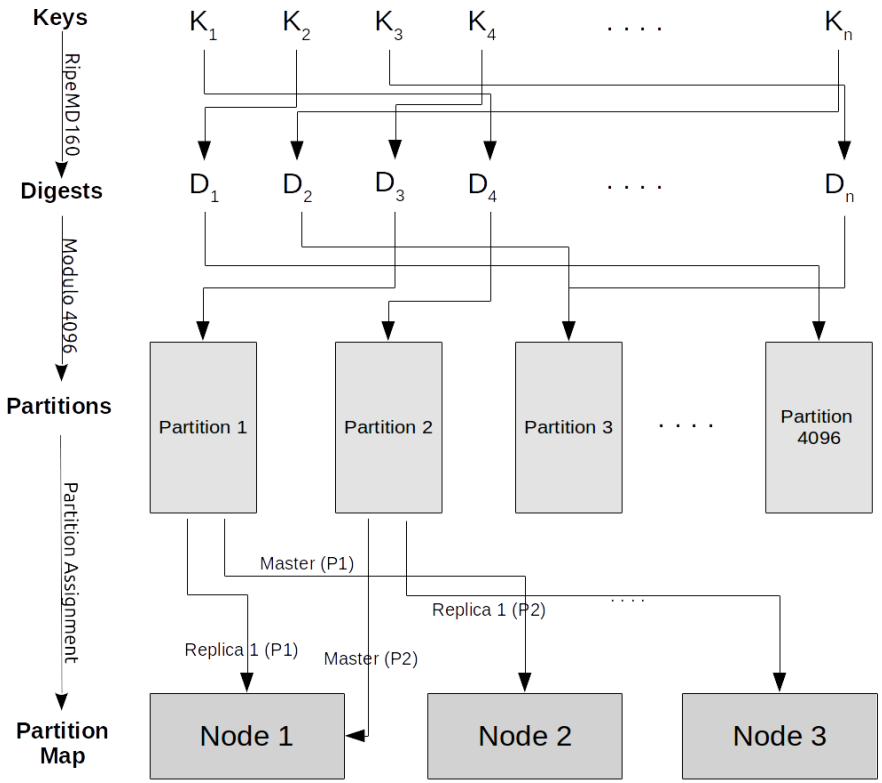
Distributed Hash Based Partitioning

- Distributed Hashing with No Hotspots
  - Every key **hashed** with **RIPEMD160** into an ultra efficient 20 byte (fixed length) string
  - Hash + additional (fixed 64 bytes) data forms **index entry** in RAM
  - **Some bits** from hash value are used to calculate the **Partition ID** (4096 partitions)
  - Partition ID maps to Node ID in the cluster



AEROSPIKE

Data Distribution



Index and data are colocated

1. Distribute workload uniformly
2. Provide predictable read/write performance
3. Scale up and down by simply adding cluster nodes
4. Rebalance data non-disruptively and efficiently

Partition assignment objectives

1. Deterministic, so each node can operate by itself
2. Uniform distribution of partitions across nodes
3. Minimize partition moves during cluster changes

AEROSPIKE

Partition Assignment Algorithm

```
function REPLICATION_LIST_ASSIGN(partitionid)
  node_hash = empty map
  for  nodeid in succession_list:
    node_hash[nodeid] = NODE_HASH_COMPUTE(nodeid, partitionid)
  replication_list = sort_ascending(node_hash using hash)
  return replication_list
```

```
function NODE_HASH_COMPUTE(nodeid, partitionid):
  nodeid_hash = fnv_1a_hash(nodeid)
  partition_hash = fnv_1a_hash(partitionid)
  return jenkins_one_at_a_time_hash(<nodeid_hash, partition_hash>)
```

Partition	Master	Replica 1	Replica 2	Unused	Unused
P1	N5	N1	N3	N2	N4
P2	N2	N4	N5	N3	N1
P3	N1	N3	N2	N5	N4
...	...	...	...	...	...

(a) Partition assignment with replication factor 3

P2	N2	N4	N3	N1	

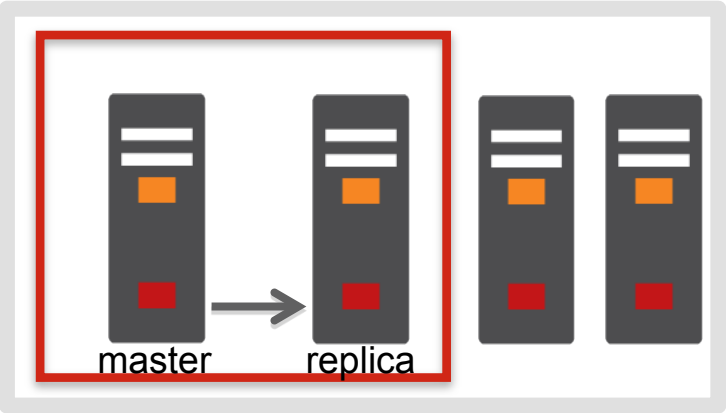
(b) P2 succession list when N5 goes down

P2	N2	N4	N5	N3	N1

(c) P2 succession list when N5 comes up again

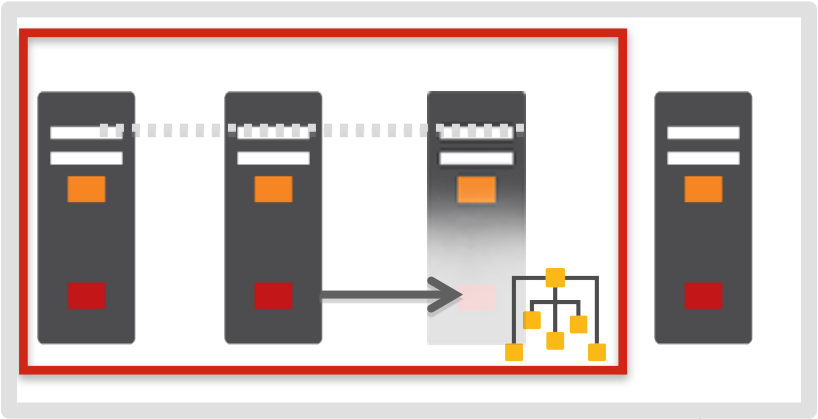
AEROSPIKE

Real-Time Prioritization



Writing with Immediate Consistency

1. Write sent to row master
2. Latch against simultaneous writes
3. Apply write to master and replica synchronously
4. Queue operations to disk
5. Signal completed transaction
6. Master merges duplicate copies (if any)



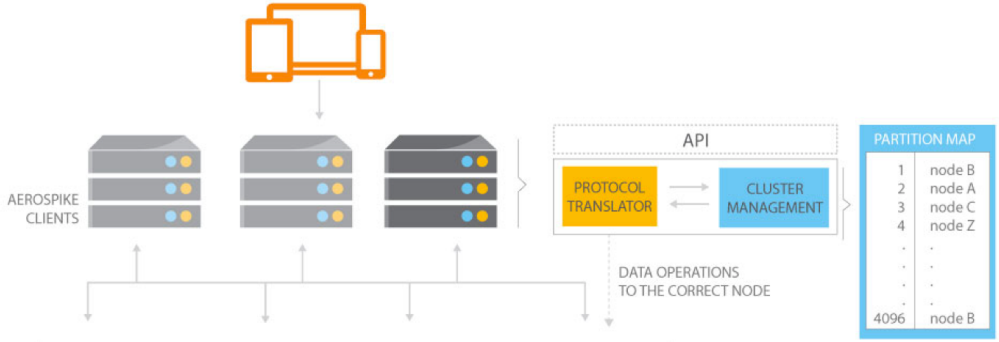
transactions continue

Adding a Node

1. Cluster discovers new node via gossip protocol
2. Paxos vote determines new data organization
3. Partition migrations scheduled (only deltas copied)
4. When a partition migration starts, write journal starts on destination
5. Partition moves atomically
6. Journal is applied and source data deleted

AEROSPIKE

Intelligent Client

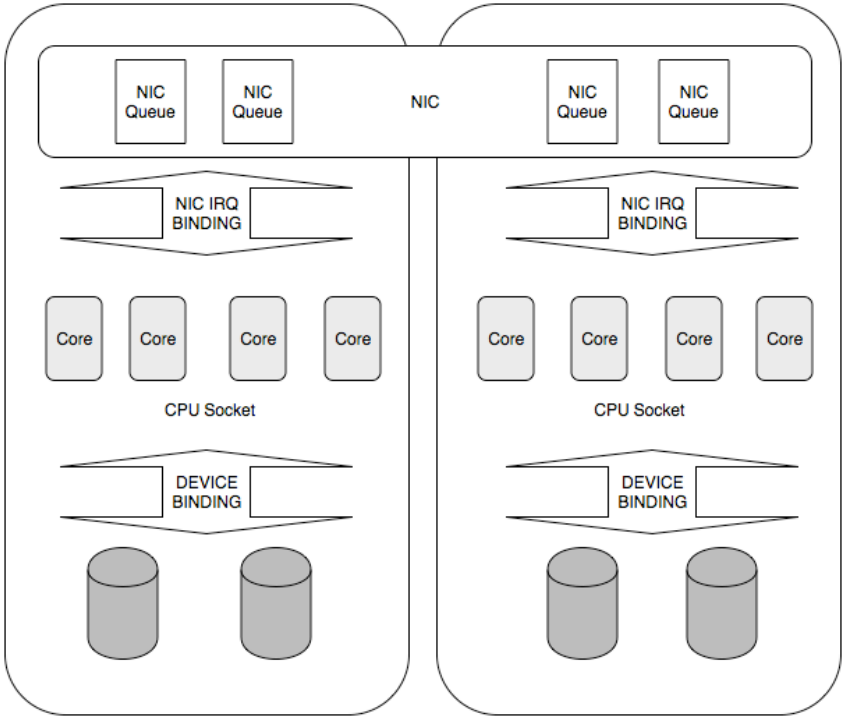


- The Aerospike Client is implemented as a **library**, JAR or DLL, and consists of 2 parts:
  - Operation APIs – These are the operations that you can execute on the cluster – **CRUD+** etc.
  - First class **observer** of the Cluster – Monitoring the state of each node and aware of new nodes or node failures.
- **1 Hop** to data
  - Smart Client simply calculates Partition ID to determine Node ID
  - Client performs load balancing

AEROSPIKE

Designed for Wire-Line Speed

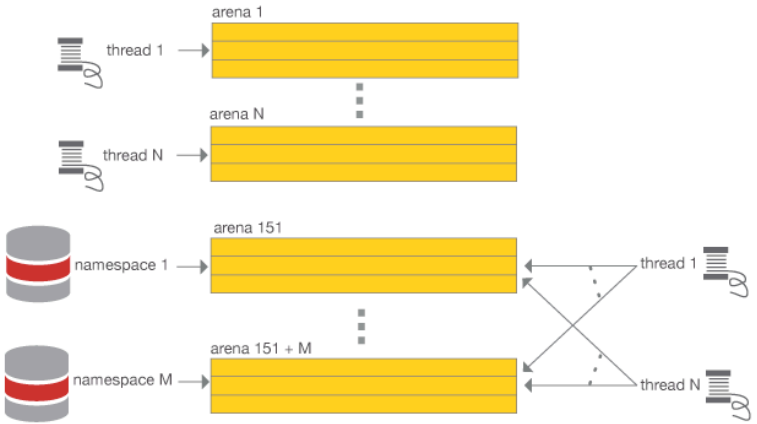
Multi-core architecture



Optimized C based DB kernel

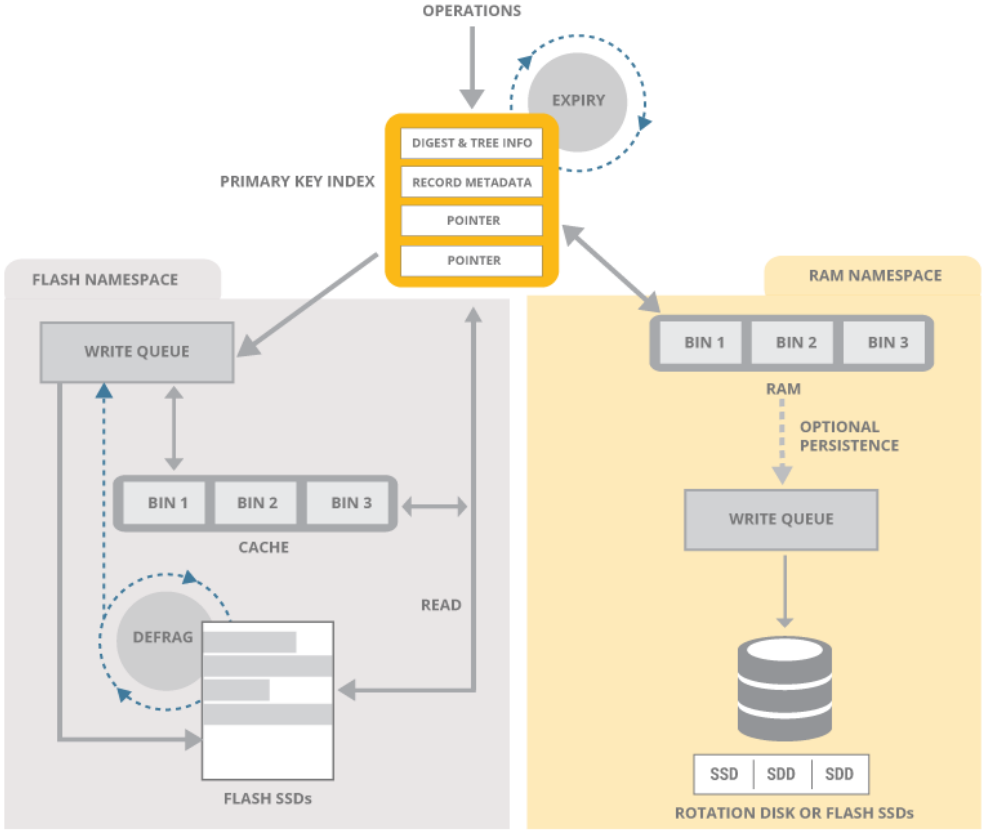
1. Multi-threaded data structures
2. Nested locking model for synchronization
3. Lockless data structures
4. Partitioned single threaded data structures
5. Index entries are aligned to cache line (64 bytes)
6. Custom memory management (arenas)

Memory Arena Assignment



AEROSPIKE

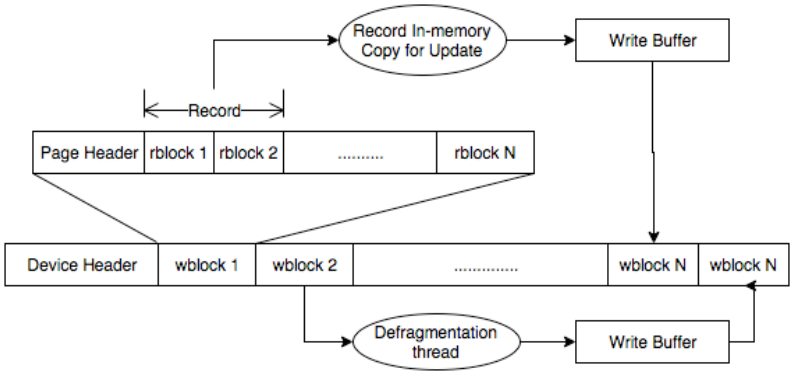
Storage Architecture



Highlights

1. Direct device access
2. Large Block Writes
3. Indexes in DRAM
4. Highly Parallelized
5. Log-structured FS “copy-on-write”
6. Fast restart with shared memory

Storage Layout



AEROSPIKE

Benchmarks

AEROSPIKE

Aerospike vs Cassandra (2016)

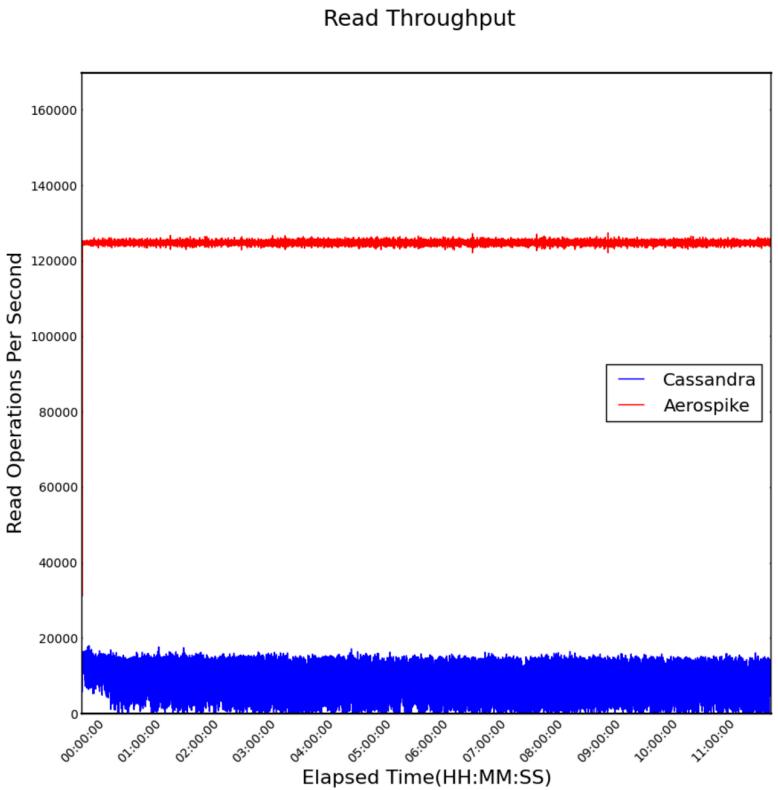
	Read Throughput (transactions per second)	Update Throughput (transactions per second)	95th Percentile Read Latency (milliseconds)	95th Percentile Update Latency (milliseconds)
Aerospike	125,000	125,000	2.3	4.0
Cassandra	8,900	8,900	97.5	94.0
Ratio	14x better	14x better	42x better	24x better

Table 1. Summary of Results

- 3 node cluster, Intel S3700 SSDs
- Followed religiously all DataStax recommendations
- Standard YCSB, includes instructions to reproduce for your workload
- http://www.aerospike.com/blog/comparing-nosql-databases-aerospike-and-cassandra/

AEROSPIKE

Aerospike vs Cassandra (2016)



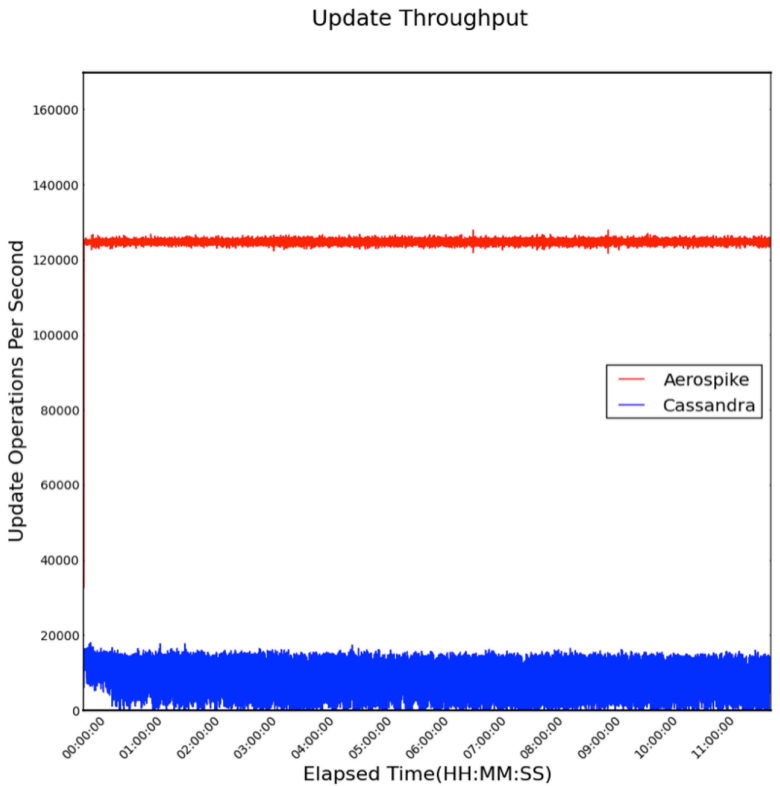


Figure 6. Update Throughput (higher is better)

Aerospike vs Cassandra (2016)

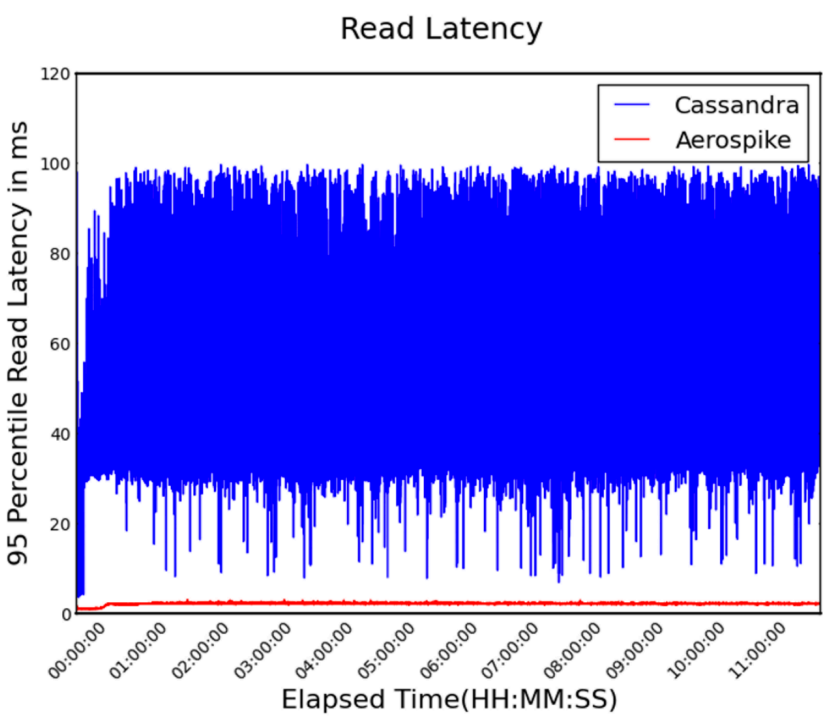


Figure 5. 95th Percentile Read Latency (lower is better)

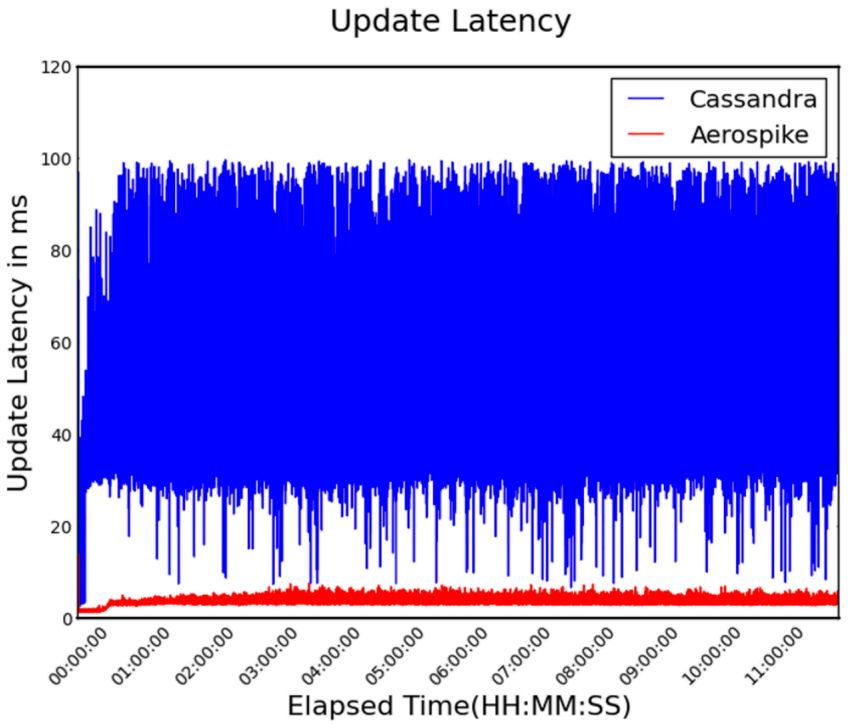
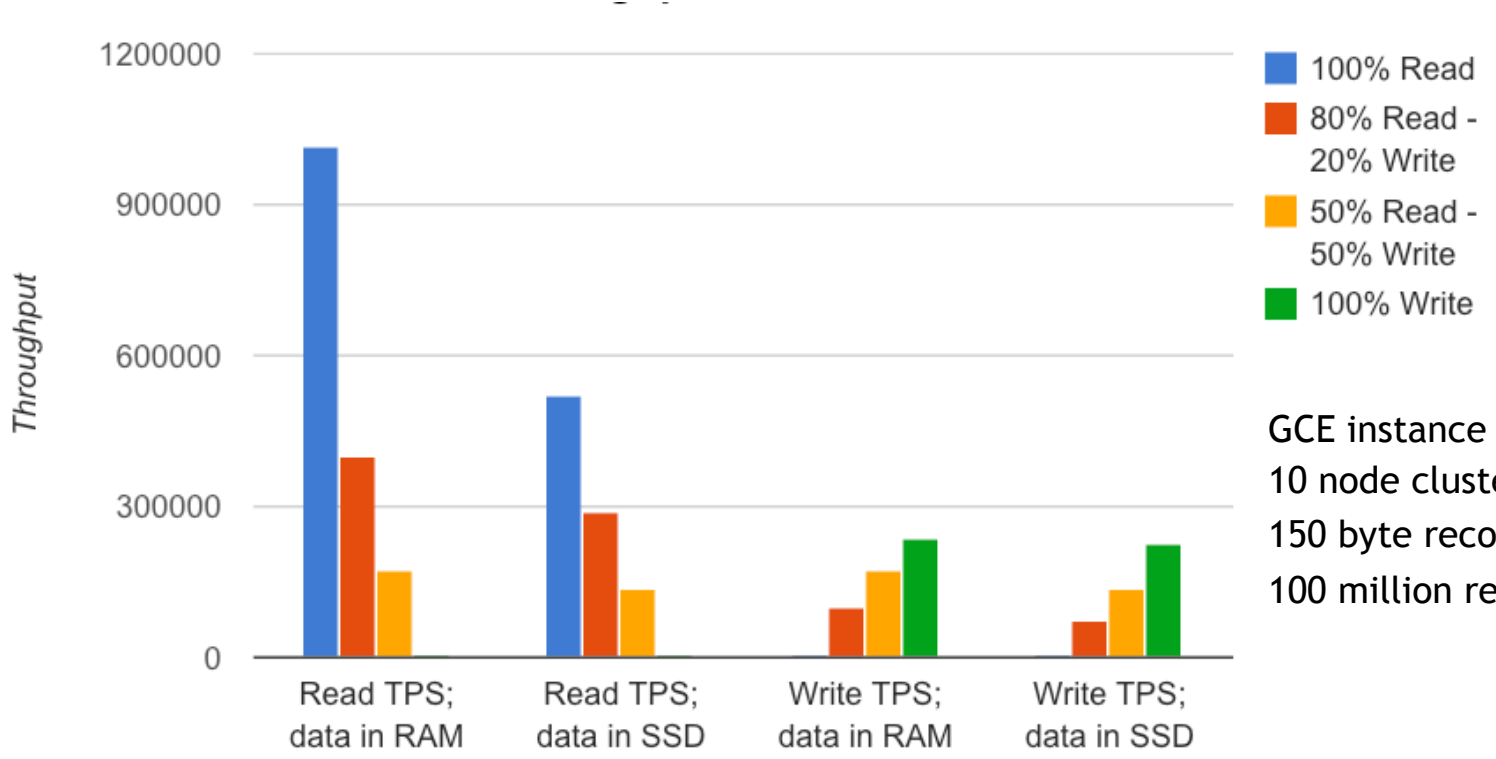


Figure 7. 95th Percentile Update Latency (lower is better)

DRAM vs SSD on GCE

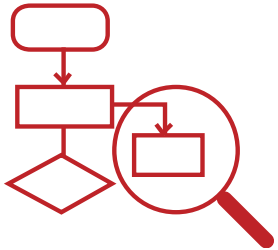


GCE instance n1-standard-8
10 node cluster
150 byte record with 3 columns
100 million records

AEROSPIKE

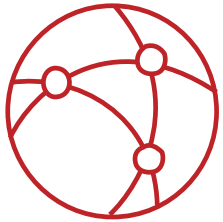
Future Work

Software



- Add CP mode
- Conflict detection and resolution
- Pipelined execution of client transactions
- Security

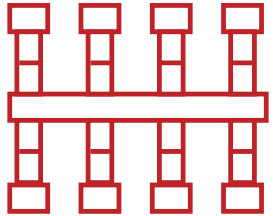
Application Requirements



Customers demand

- Real-time decisions based on recent data
- High Consistency
- Security

New Hardware



- 3D XPoint
- High core CPUs
- NVMe
- Multi-queue network cards
- Virtualized IO

AEROSPIKE

Bonus: Flash and Storage Class Memory

AEROSPIKE

NVMe Arrives – 2015 to present

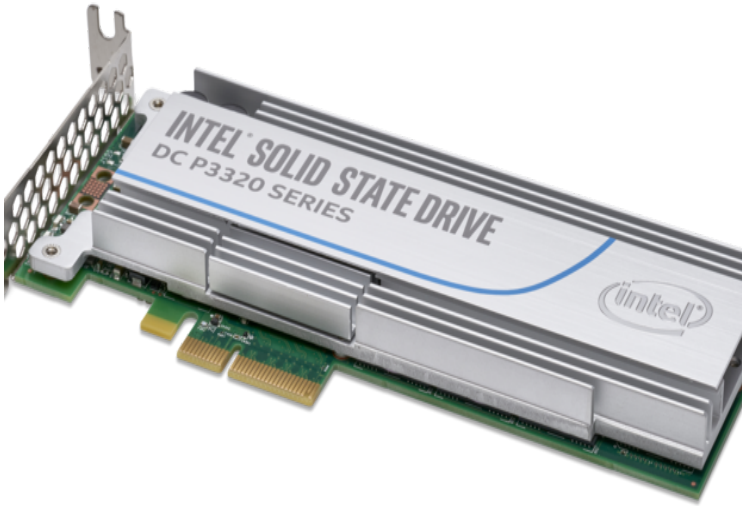
- **Linux, Windows drivers achieve performance**
- **U.2 and M.2 form factors available**

- **Intel P3700, P3600, P3500** available –
  - 250k IOPs per card
- Samsung PM1735 available –
  - 120k IOPs per card
- Micron 9100
- HGST, Toshiba – 30k to 50k per card

- **SAS / SATA lingers**
  - Samsung SM1635, PM1633; Intel S3700; Micron S600 still shipping

NVMe's crushing superiority

- **High, predictable performance**
  - High transfer speed reduces jitter
    - 2.8uS NVMe vs 5.0uS SATA
  - Better controllers
  - Mature and tuned Linux driver
  - U.2 front panel hot swap available (and 24-wide)
- NVMe arrays available
  - Apeiron ADS1000 "direct scale-out flash"
  - EMC D5, Mangstore NX63020
- All new Aerospike deployments on NVMe



Flash in the Public Cloud

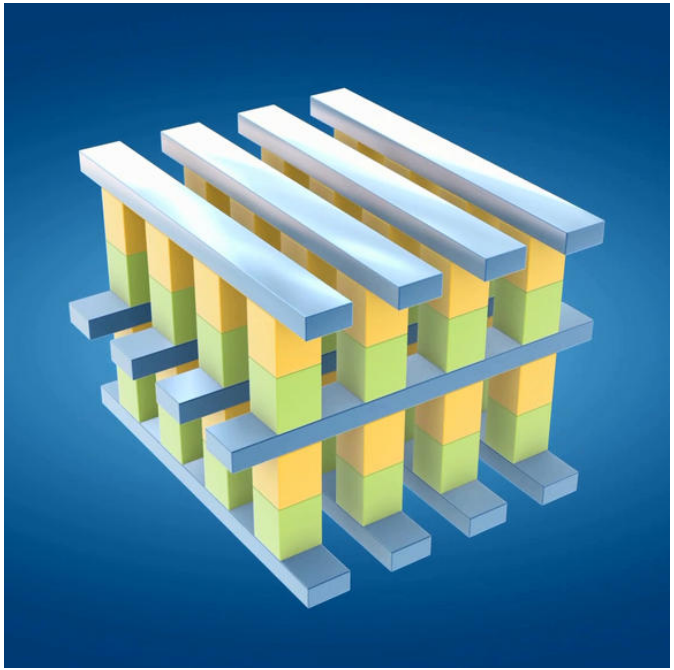
- **Every public cloud provider has Flash**
  - AWS / EC2 has sophisticated offerings
  - Google Compute is high performance
  - Softlayer allows own-hardware
- Private clouds manage Flash
  - Docker offers storage metadata
  - Pivotal manages Flash and traditional storage

“All Flash arrays”

- Database knows best, not storage
  - Database should manage
    consistency vs availability
  - Database should manage views and snapshots
- Array vendors have started making “databases”
  - “Object stores”
- High Density “flash aggregation”
  - Sandisk Infiniflash – SATA
    - High-read, or write-once, applications
  - Apeiron ASD1000 – NVMe
    - Read and write applications
  - Vexata, others

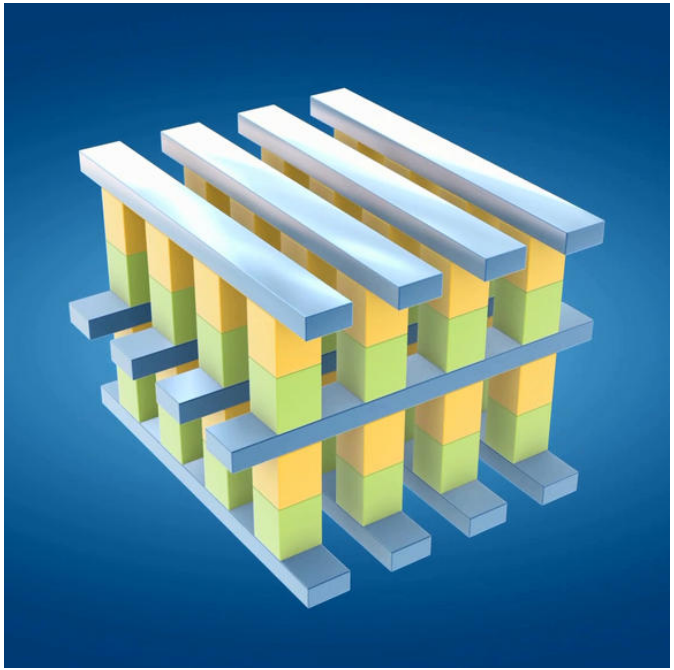
What is 3D Xpoint?

- Persistent storage using chips
  - No power while idle
- *Does not use transistors*
  - Resistor / phase change “but different”
- Chips almost as fast as DRAM (100 ns, not 5ns)
  - NAND is 10 ms to 5 ms
- Higher density than DRAM, lower than NAND
- “Infinite” write durability
- 128B read and write granularity
  - NAND write granularity --- 16 MB
  - DRAM write granularity --- 64 B



Intel's 3D Xpoint roadmap (public info)

- Optane *this year*
  - 3D Xpoint in 2.5" NVMe package
  - "7x faster" – limited by NVMe !
- In the public cloud NOW
  - BlueMix announcement
  - Others to come
- NVDIMM (on memory bus)
  - Removes NVMe limit
  - Intel cagy on delivery – "uncommitted"
  - *Hard to program to*



Thank You

Questions?

brian@aerospike.com

AEROSPIKE