Aerospike : High Performance NoSQL Store with Flash Optimization

Gagan Agrawal, Sr. Principal Engineer, Snapdeal

Agenda

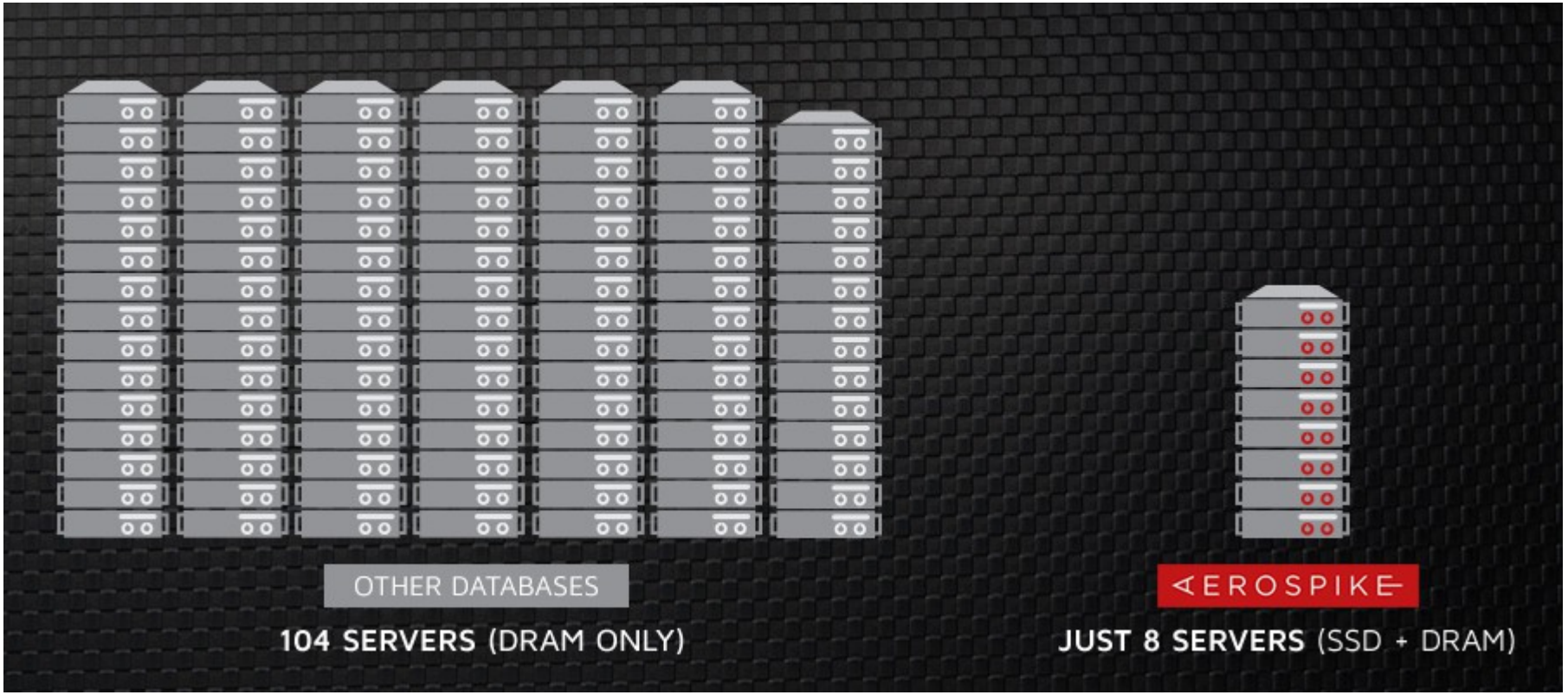
- What is Aerospike
- Aerospike Architecture
- Data Model
- Demo
- Aggregation
- Aerospike @ Snapdeal

What is Aerospike ?

Aerospike

- Distributed and Scalable Key Value Store
- Designed for Flash Optimized Storage
- 99% < 1 millisecond
- 1 M TPS / 100 TB
- 100% Uptime with strong consistency (ACID)

Comparison

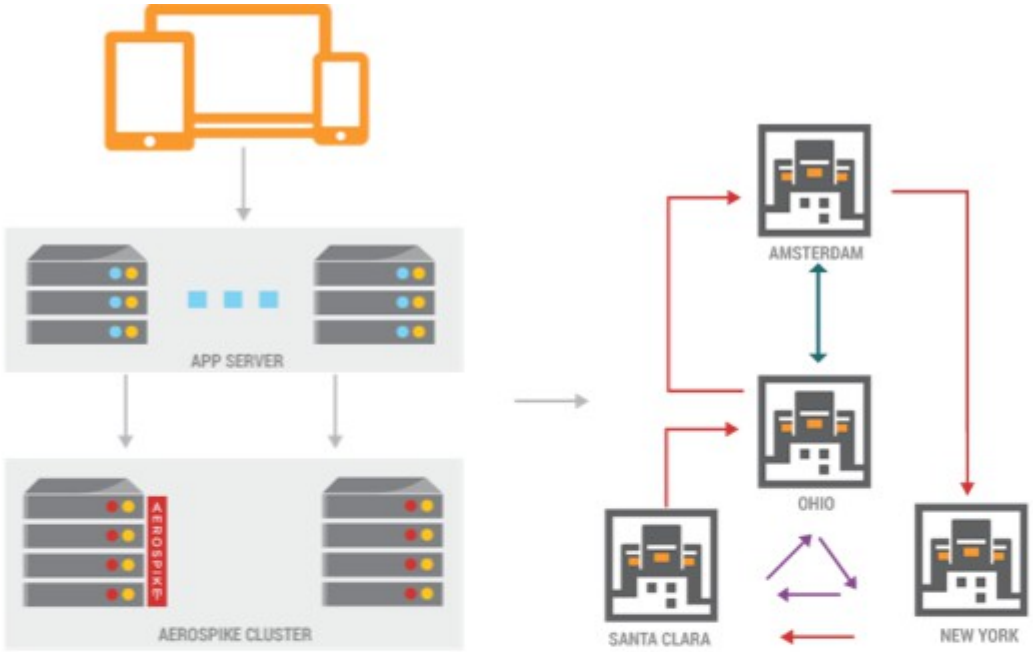

OTHER DATABASES
104 SERVERS (DRAM ONLY)
AEROSPIKE
JUST 8 SERVERS (SSD + DRAM)

Aerospike Architecture

Architecture Objectives

- To create flexible and scalable platform
- To provide robustness and reliability (ACID)
- To provide Operational Efficiency

Architecture



1) **No Hotspots** – DHT simplifies data partitioning

2) Smart Client – **1 hop** to data, no load balancers

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

4) Single row **ACID** – synch replication in cluster

5) Smart Cluster, **Zero Touch** – auto-failover, rebalancing, rack aware, rolling upgrades

6) Transactions and long-running tasks prioritized in real-time

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

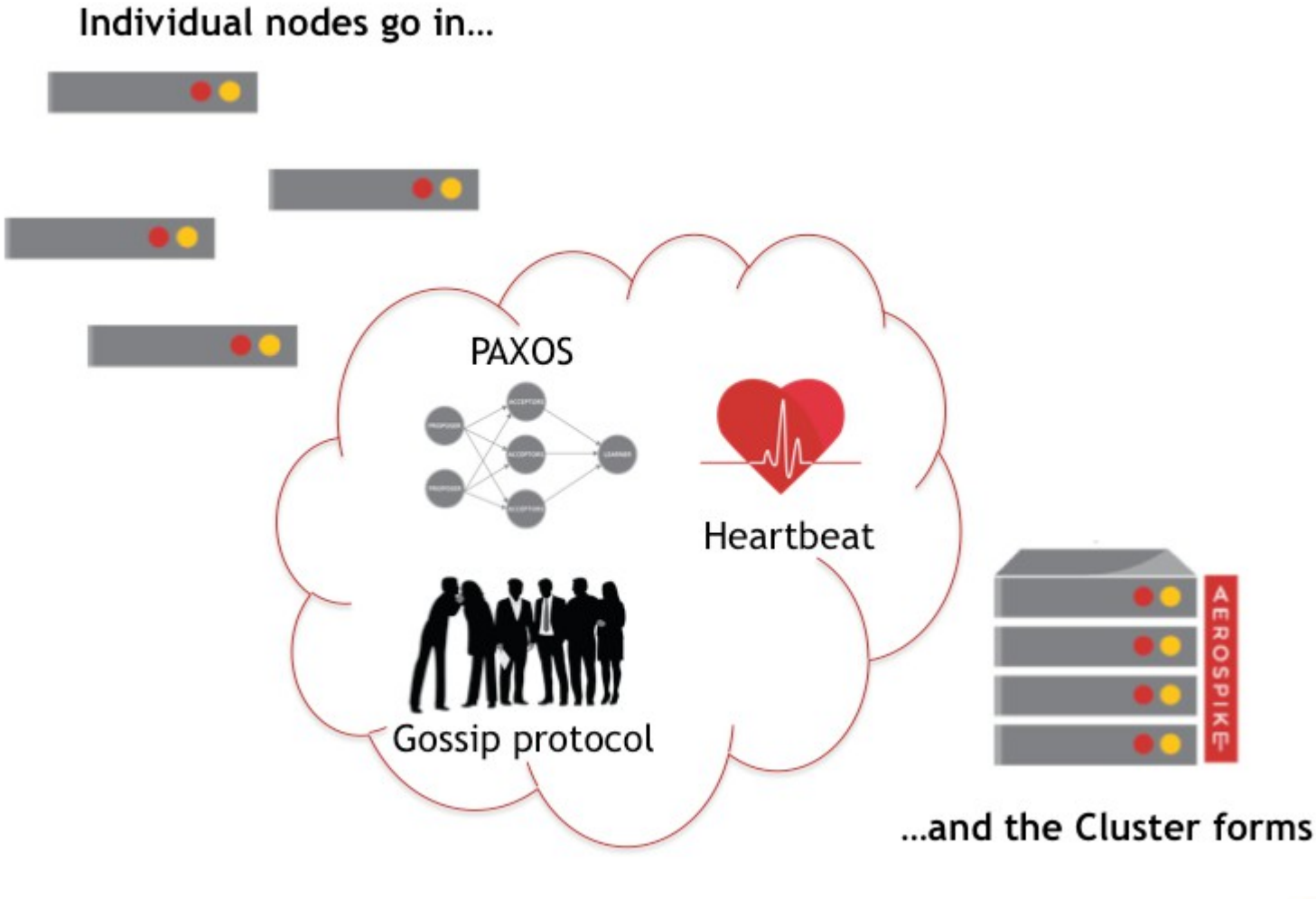
3 Layers

- Cluster aware Client Layer
  - Track Nodes
  - Know where data resides in Cluster
  - Implements its own TCP/IP connection pool
- Self Managing Clustering and Data Distribution Layer
  - Automatic fail over
  - Replication
  - Intelligent Re-balancing and data migration
- Flash Optimized Data Storage Layer
  - Stores data in RAM and Flash

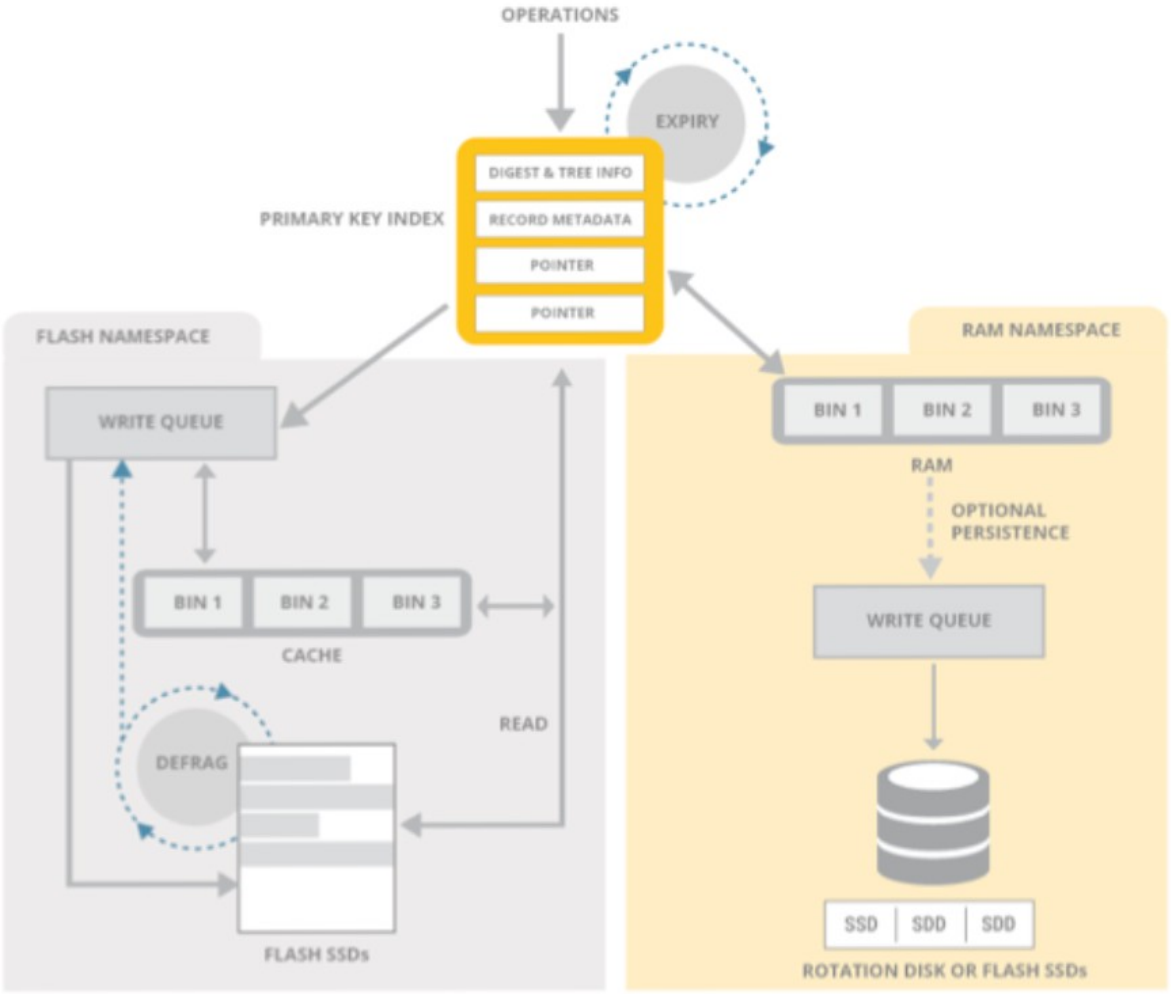
Cluster Layer

- Cluster Management Module
  - Tracks Nodes in the Cluster
  - Paxos like consensus voting process
- Data Migration Module
  - Balances distribution of data
  - Ensures data duplication as per replication factor
- Transaction Processing Module
  - Sync / Async Replication
  - Proxy
  - Duplicate Resolution

Cluster Formation



Data Storage Layer



Data Storage Layer

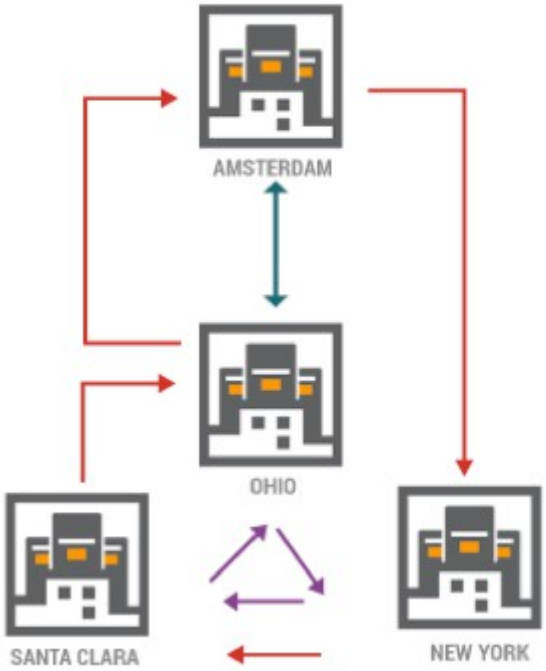
- Designed for speed
- Can operate all in-memory or with flash storage
- 100 Million keys take up only 6.GB
- Log Structured
- Writes to disk are performed in large blocks
- Bypasses standard file system
- Built in Smart Defragmenter and Intelligent Evictor

Scales on Commodity Server

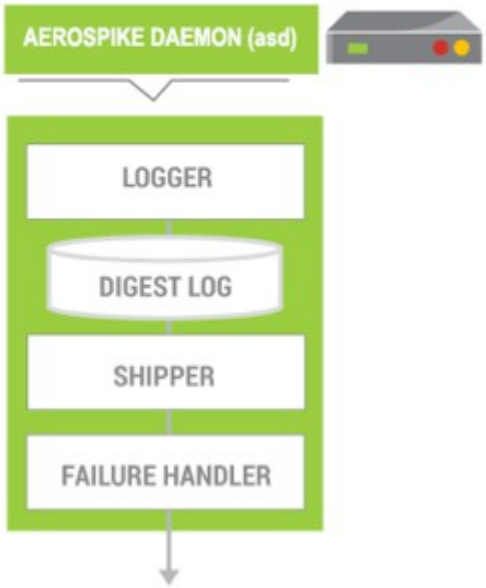
- Each server scales to manage upto 16TB data
- Parallel access to up to 20 SSDs
- Scales out linearly with each identical server
- Battle tested with 100TB+ across the cluster

XDR Architecture

Distributed clusters



Each node in the cluster

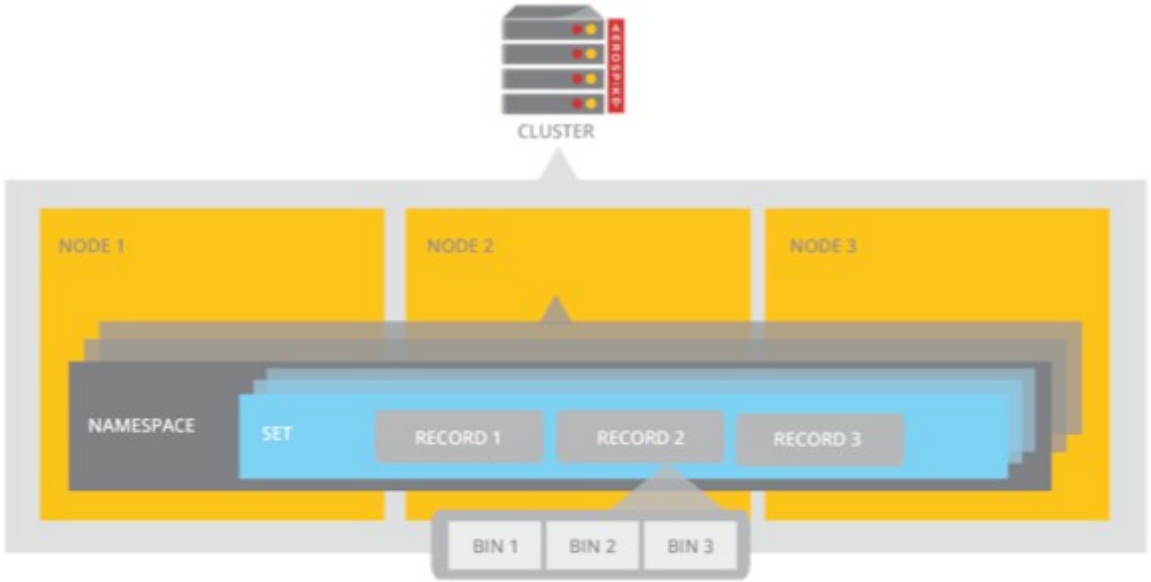


Data Model

Data Hierarchy

- Namespace (Database)
- Set (Table)
- Record (Row)
- Bins (Columns)

How Data is Organized

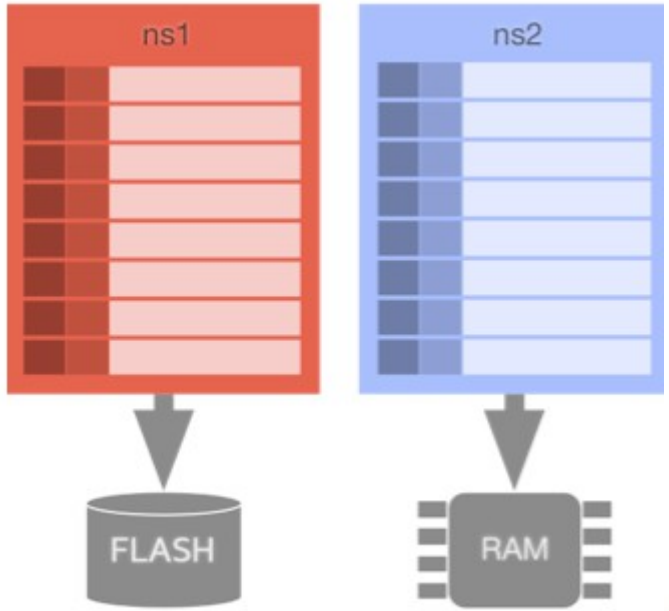


Aerospike	RDBMS
Namespace	Tablespace or Database
Set	Table
Record	Row
Bin	Column

Bin type*
Integer
String
BLOB
List
Map

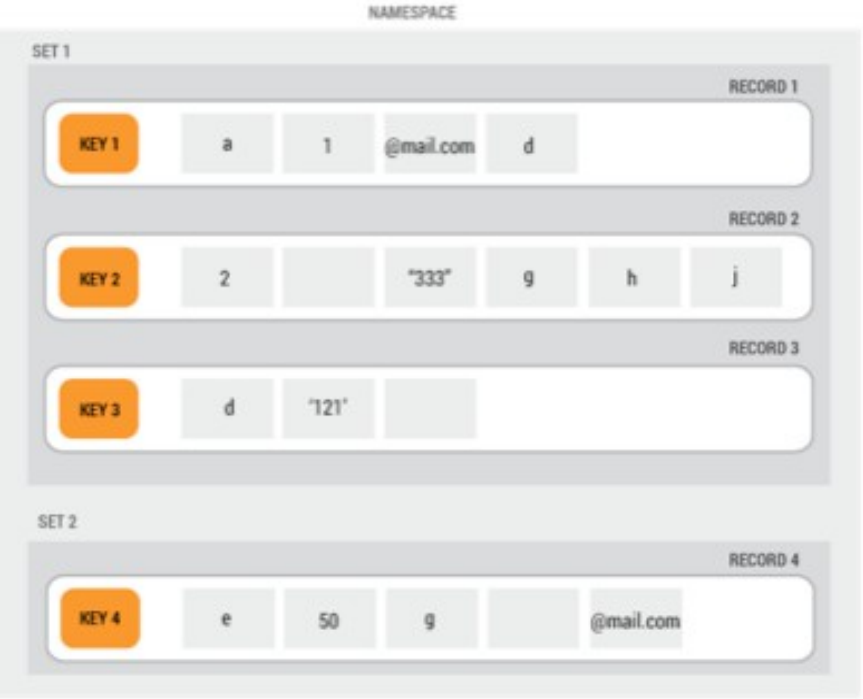
Namespace

- Storage definition- DRAM or Flash
  - Storage block size (128k – 2MB)
  - Controls size of Ram and Storage
- Policy Container
  - Replicator Factor
  - Default Expiry
- Data Container
  - Namespace contains Set
  - Set contains Records



Set

- Similar To Table
- But has no Schema
- Inherits policy from NS
- Prefix to Primary Key
- Name <= 63 chars
- 1023 per Namespace
- Cannot be deleted or renamed



Record

- A record is row of key-value
- Value : one or more bins
- Bin has a name and type
  - Types : String,Integer,Blob,List,Map, Large Data Types (LDT)
- Bins can be added at any time
- Generation Counter
  - Optimistic Concurrency
- Time-to-live
  - Auto Expiration

Bins

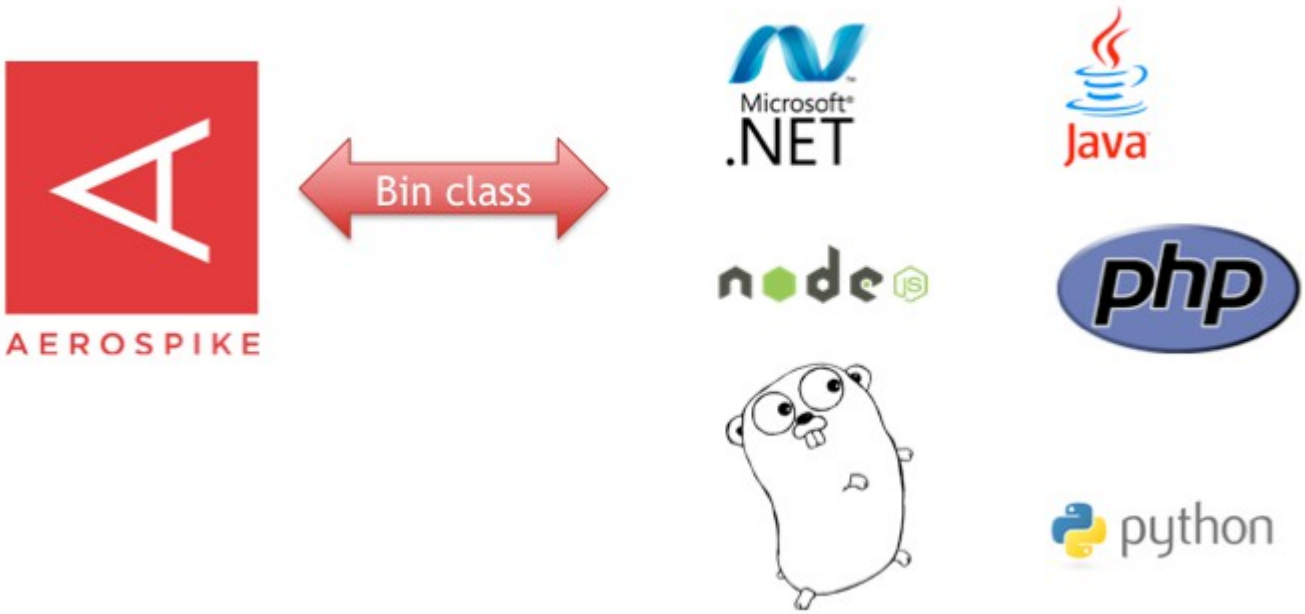
- Bins have a:
  - Name : 14 character or less
  - Type : one of following
    - String, Integer, Blob, List, Map
    - Large Data Types
      - Large Ordered List
      - Large Map
- Bins are stored in the record
- A Bin can have different type in another record

Bins

Id	lname	fname	address	favorites
1	Able	John	123 First	Cats, dogs, mice
2	Baker	Kris	234 Second	
3	Charlie			
4	Delta	Moe	456 Fourth	Stake, ice cream apples

Type Mapping

- Types are mapped to equivalent language type

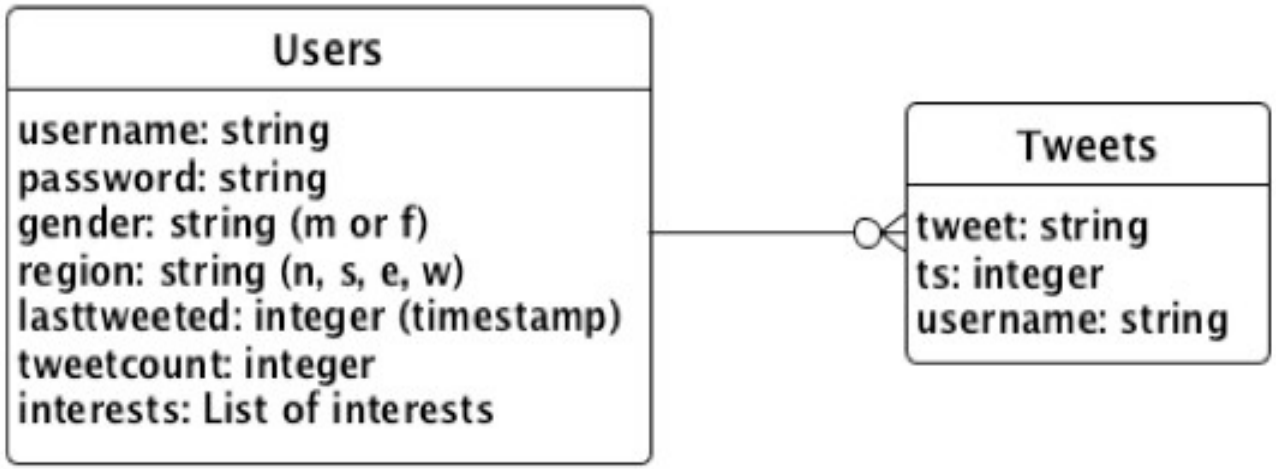


Demo and Code Samples

Data Modeling

- Focus on how you will query the data
  - All Tweets from a given user
  - Give me the last 10 Tweets for a given user
  - Last 10 Tweets for all users
  - How many users Tweeted in last X minutes

Data Modeling



Secondary Index

- Value based lookup
- Query is sent to all nodes in the cluster in parallel
- Best for low selectivity indexes
  - Result set of 1k to 1million

UDF

- Move compute close to data
- UDFs are common in many databases
  - MySQL
  - SQL Server
  - Oracle
  - DB2

UDF

- Written in Lua
- Record oriented
  - acts on a single record (row)
- Stream oriented
  - Acts on a stream of records resulting from a Query
- Using UDFs you can move processing to the same node as the data

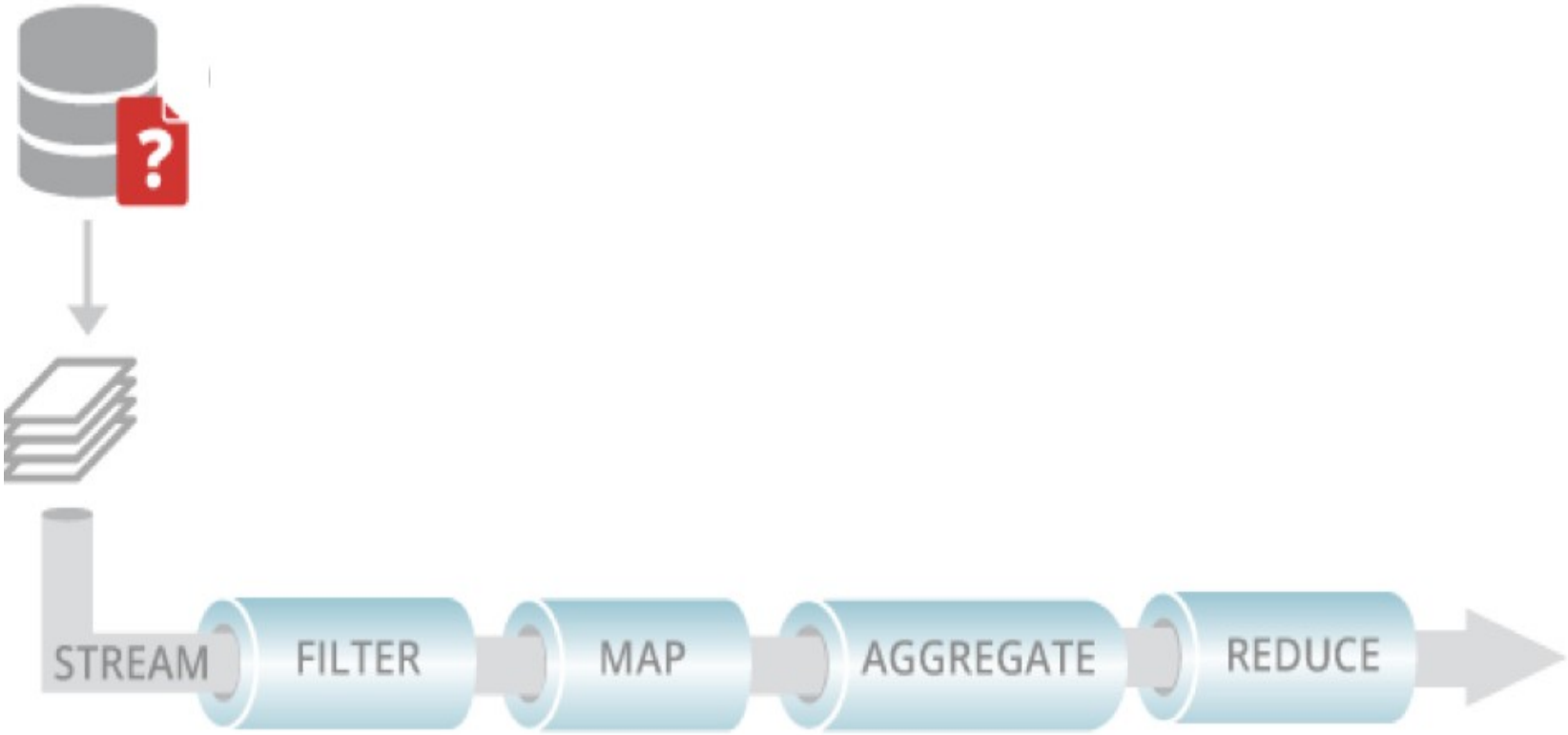
Example Lua UDF

Aggregation Framework

Aggregations

- Programmatic framework similar to Map Reduce
- Processes a collection of rows
- Primarily used for counts, aggregate and sums
- Runs in parallel on all cluster nodes
- Lua for function language
- Reduce in the client

Aggregations



Aerospike @ Snapdeal

Thank you!