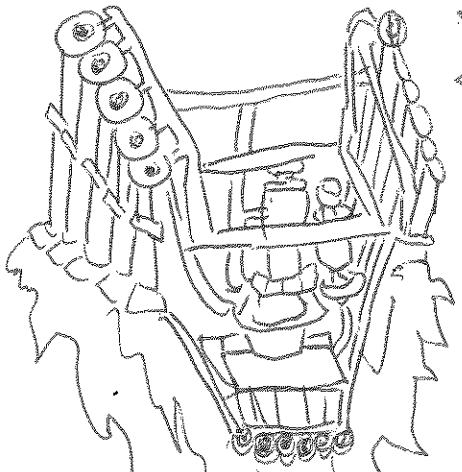
Hear what Aerospike's founders have to say:

"Aerospike: Architecture of a real-time operational DBMS" 2016.

I've only scraped the surface!

AEROSPIKE



A distributed database zine

By Sherri Li

Data Distribution
consistency
Clustering
Replication
Partitioning
Sharding
NO SCHEMA
concurrency
Data Migration
Reduce
Paxos
Passive
Active
what is it?
Gossip
Heartbeat
Transaction Processing
Inter. node Connectivity

A BIT ABOUT SHERRI:

I'm currently a fourth-year student at UCSC

studying Math as well as computer science

and I love understanding how systems work!

Sometimes I use social media:

@ricegrill

@cheesfrog

shcli@ucsc.edu

Thanks for reading!

Some sources I would be lost without them:

- Julia Evans jvns.ca
- Aerospike.com/docs/
- compositional Lindsey Kuper
- "selecting the right DB for the right job" (Bak, 2014).
- "Aerospike Beats out Cassandra, Couchbase, MongoDB: Handles Node Failure Like a Champ" (Deutscher, 2013).

For additional intellectual cravings:

- Checkout Aerospike's GitHub
- Read about their Cross-Data-Center replication, which is kind of like chain replication
- "On the collision resistance of RIPEMD-160" 2006.

within a cluster, a node can fail. Other nodes track one anothers status!

♡This protocol is called heartbeat♡

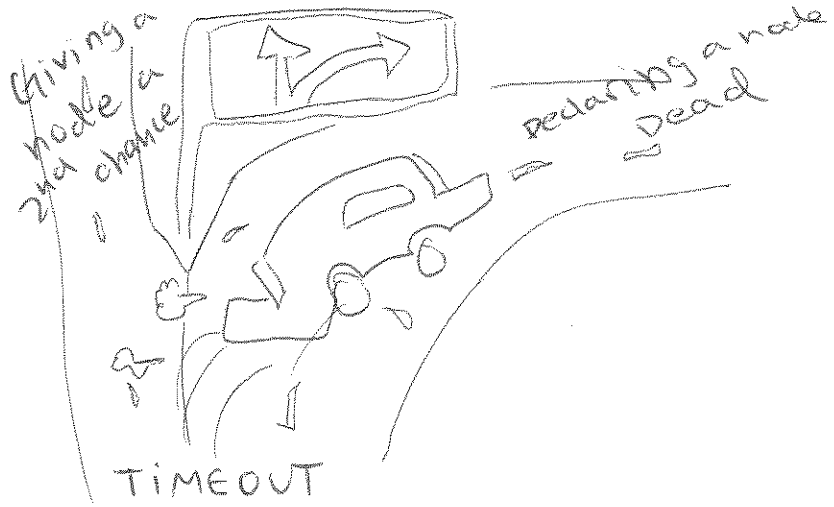
You can setup TCP unicasting, where each node "pings" other nodes:

```
heartbeat {
    node mesh
    address <IP>
    port <port # on which the nodes is listening>
    mesh-seed address-port <IP for a node in the cluster>
    .
    .
    .
    interval 150
    timeout 10
}
```

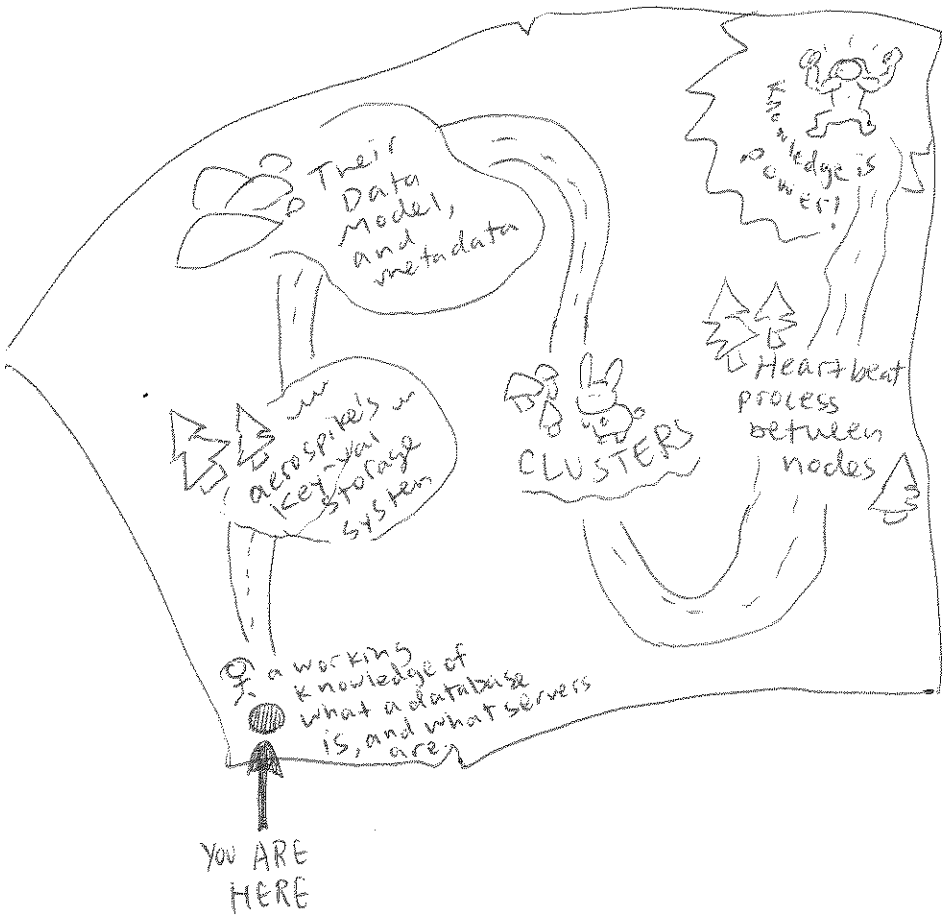
Or you can Set up Multicasting

```
>heartbeat{
    node multicast
    multicast-group <IP>
    port <port #>

    interval <# millisec. between heartbeats>
    timeout <before declaring a node "dead">
}
```



Here's a roadmap of what you're about to EXPLORE today:



Let's begin our journey!

What is Aerospike?

It's a type of rocket engine that maintains aerodynamic efficiency across a wide range of altitudes, by using a nozzle that forms an "air spike."

Wait.... You thought we were talking about databases?

Oh right.

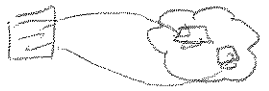
AEROSPIKE is a company that produces the Aerospike database, a flash-optimized in-memory open source NoSQL D.B.!

How is data distributed?

→ All nodes in a cluster are peers

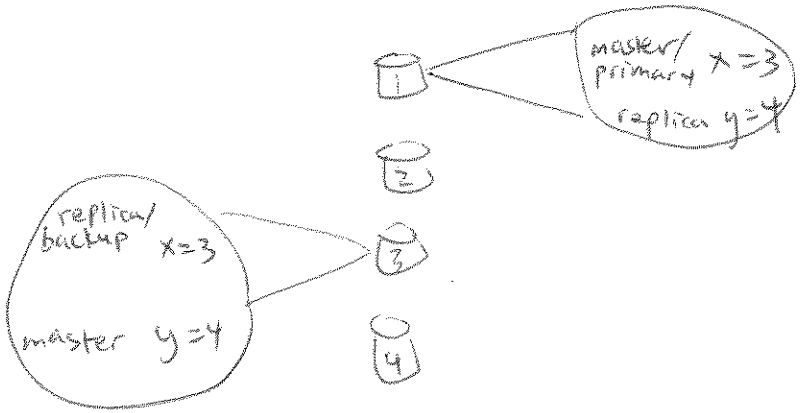
there is no "master."
(a "shared-Nothing" architecture)

→ The RIPEMD-160 evenly distributs data across all nodes in the cluster



→ Each "cluster," or group of nodes, stores *master* data and *replica* data!

Consider a 4-node cluster:



Each node is a "master" of $\frac{1}{4}^{th}$ of the data, and a "replica" of another $\frac{1}{4}^{th}$.

Now, what about each record's metadata?

The METADATA consists of:

1. Generation#: to track the latest modifications made to the record, the total # of modifications.
2. Time To Live (TTL): specify the record's expiration.
3. Last_Update-Time (LUT): specify the latest update time, a timestamp.

if a node goes down and a write, you take the record with the higher generation#, or the record with the most recent LUT!

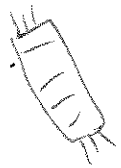
And bins?

Each consists of a name and a value.

What does the jargon actually mean though?

1. Flash-optimized:

Flash memory = stores data in an array of memory cells



→ no moving parts leads to faster access to data and more durability than hard drives

2. NoSQL:

SQL:  a database structured to recognize relations among items.

T1

name	student ID	major
Brian	1	Finance
Srini	2	Math
Psi	3	Physics

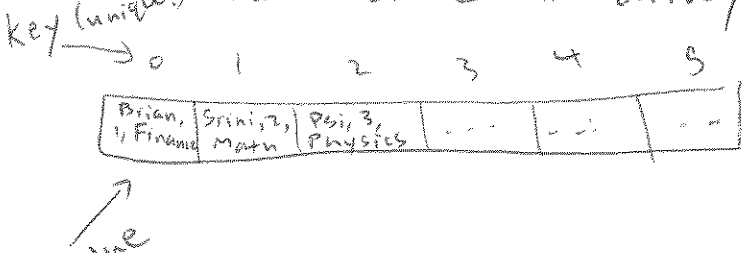
T2

student ID	Dorm
1	college ID
2	Merrill
3	Kresge

The "key" student ID links T1 and T2.

→ No-SQL databases are often "key-value" stores.

- No schema
- data stored in arrays



- faster reads and writes.

→ The path to retrieve the data is an O(1) operation: A direct request to the object in memory.

Hashing the keys into smaller digest values helps with minimizing storage space!

Rather than storing

key=(students, names, Hmmhmmmmmmmmm)

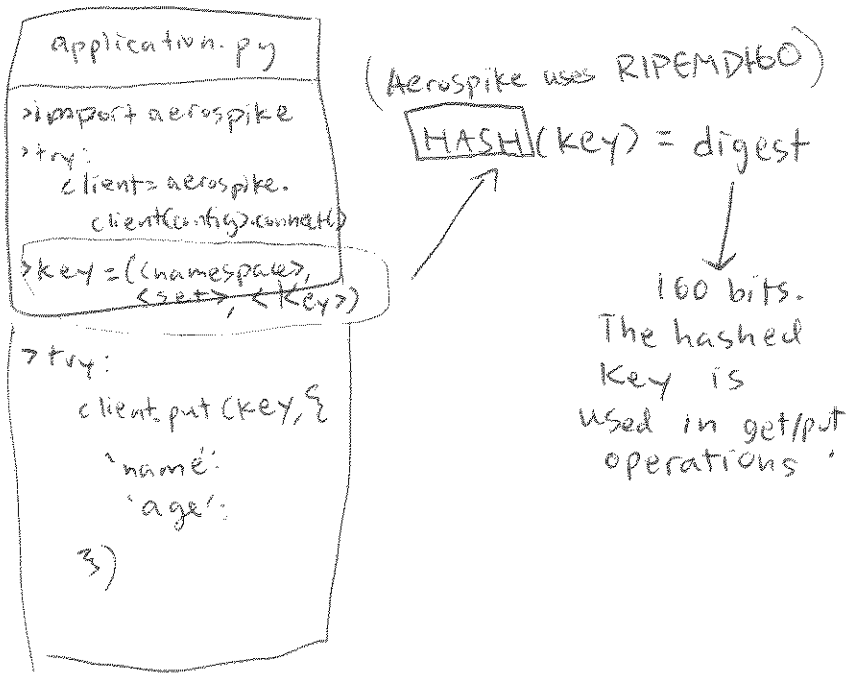
we can store some value like

DCED 3544 B8AC 987A 75A1 696C D429 6709 C191 FLCA

instead. It comes in handy when your keys get longer!

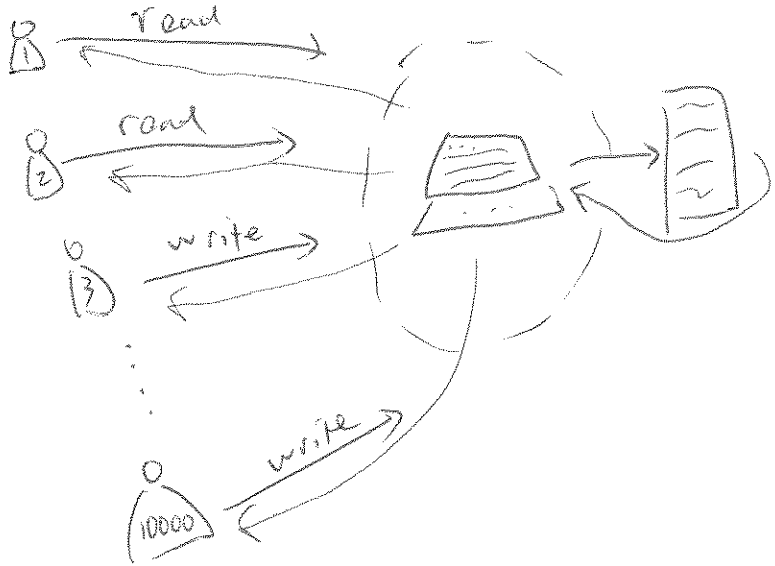
→ a record contains:

1. key: unique identifier
2. metadata: version info, and expiration
3. bins: a value corresponding to the data type of the record.



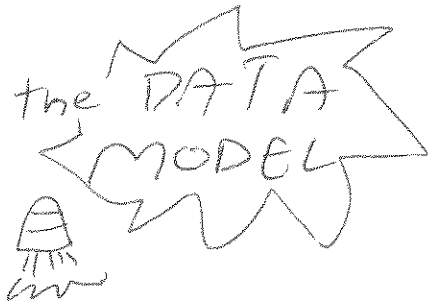
Why use a NoSQL key-val store?

1. They can handle a higher stream of operations, with lower latency!



2. You can scale out using partitioning, or storing data on multiple nodes.

Let's get into the DATA MODEL of Aerospike



- a "namespace" is a container for your ~~data~~, just like how Docker is a container for your application.

> namespace students {

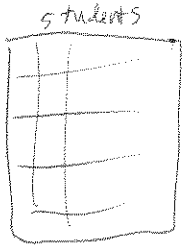
. . . .

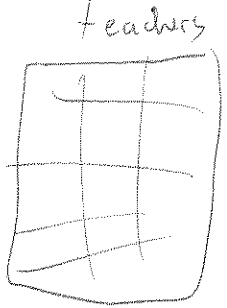
}

> namespace teachers {

. . . .

}

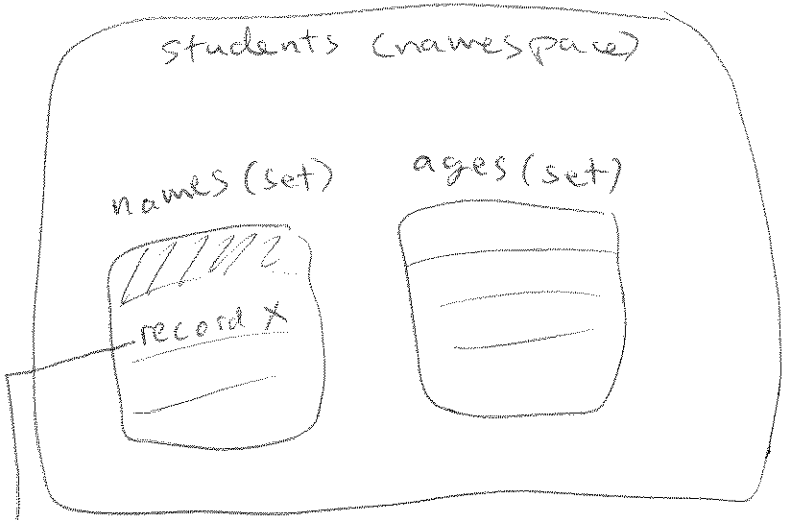




namespaces bind your data to a storage device, such as a RAM segment, disk, or a file.

→ each namespace is divided into 4096 logical partitions!

- a "set" is a container within your namespace. Each set contains similar records.



- a "record" is a basic unit of storage.