Riak

Sandeep Jagtap
Amar Potghan

ThoughtWorks Inc

What

- dynamo based
- open sourced
- key/value datastore
- scale predictably and easily

An open sourced dynamo based data store built to scale predictably and easily

Relational To Riak

Oracle	Riak
database instance	Riak cluster
table	bucket
row	key-value
rowid	key

Why

- Data doesn't fit on single server (need of distribution)
- **Downtime is unacceptable**

Why not

- You don't know that you need a distributed database
- Relationship among the data
- Query by data
- Operations on Multiple keys
- Performance > Availability

Features

- Availability
- Scalability
- Operational Simplicity
- Fault Tolerance

Availability

- Masterless, every node can service client
- Fallbacks are used when nodes are down
- Always accepts read/write requests

Scalability

- Default configuration is cluster
- Linear improvement in the performance when more nodes are added

Operational Simplicity

- Built for web
- HTTP/REST
- Add nodes easily without downtime

Fault Tolerance

- All nodes participate equally
- All data is replicated
- Cluster transparently survives:
  - Node failure
  - Network Partitioning

cluster?

- Connected computers that can be viewed as one system!

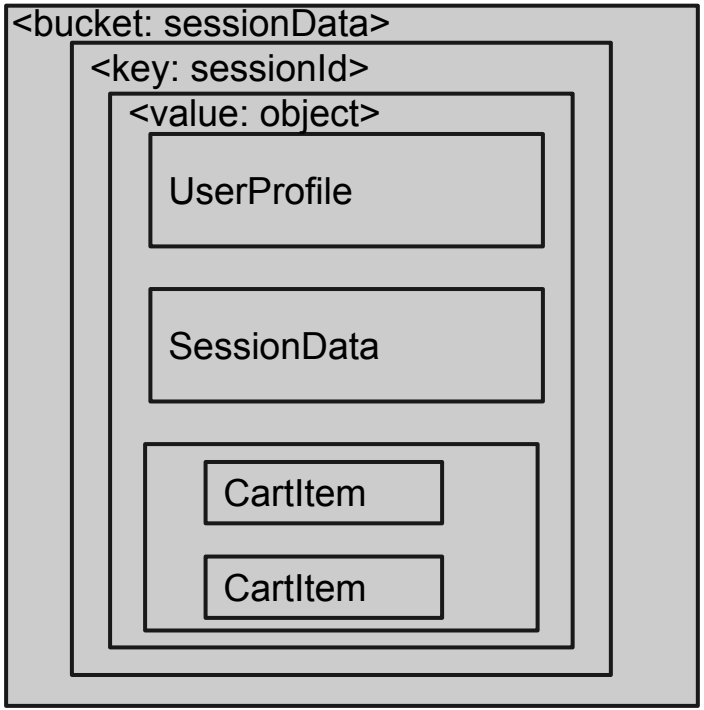
Why cluster?

- more computers -> faster processing
- can add nodes as system grows (scalability) improves
- system works even if one/two/three/N-1 go down (Availability)
- Increasing processor speed no more possible

Concepts

- Buckets
- Keys
- Values

Example


<bucket: sessionData>
<key: sessionId>
<value: object>
UserProfile
SessionData
CartItem
CartItem

Let's play with Riak

Start Single Riak Node

```
> dev/dev1/bin/riak start
```

Lets Ping Riak

> dev/dev1/bin/riak ping

pong

Lets go database way!

- Insert
- Delete
- Update
- Select

Insert (POST)

- Lets add some buckets/keys
  - curl -v -XPOST -d 'ABCDEFGHIJKLM' -H "Content-Type: application/json" http://localhost:10018/buckets/user-sessions/keys/sandeep?returnbody=true | format-json
  - curl -v -XPOST -d '{"cart":{"books":[{"name":"nosql","quantity":10}]}}' -H "Content-Type: application/json" http://localhost:10018/buckets/shop-carts/keys/pramod?returnbody=true | format-json

List Keys (GET)

- `curl -v http://localhost:10018/buckets/user-sessions/keys?keys=true |format-json`
- `curl -v http://localhost:10018/buckets/shop-carts/keys?keys=true |format-json`

List Values

- `curl -v http://127.0.0.1:10018/buckets/user-sessions/keys/sandeep`
- `curl -v http://127.0.0.1:10018/buckets/shop-carts/keys/pramod`

Update (PUT)

- Lets add some buckets/keys
  - curl -v -XPUT -d '{"cart":{"books":[{"name":"nosql","quantity":11}]}}' -H "Content-Type: application/json" http://localhost:10018/buckets/shop-carts/keys/pramod?returnbody=true | format-json

Delete (DELETE)

- Lets delete some Key
  - `curl -v -X DELETE http://127.0.0.1:10018/buckets/user-sessions/keys/sandeep`
- Lets see if they are deleted
  - `curl -v http://127.0.0.1:10018/buckets/user-sessions/keys/sandeep`

Start 5 node cluster

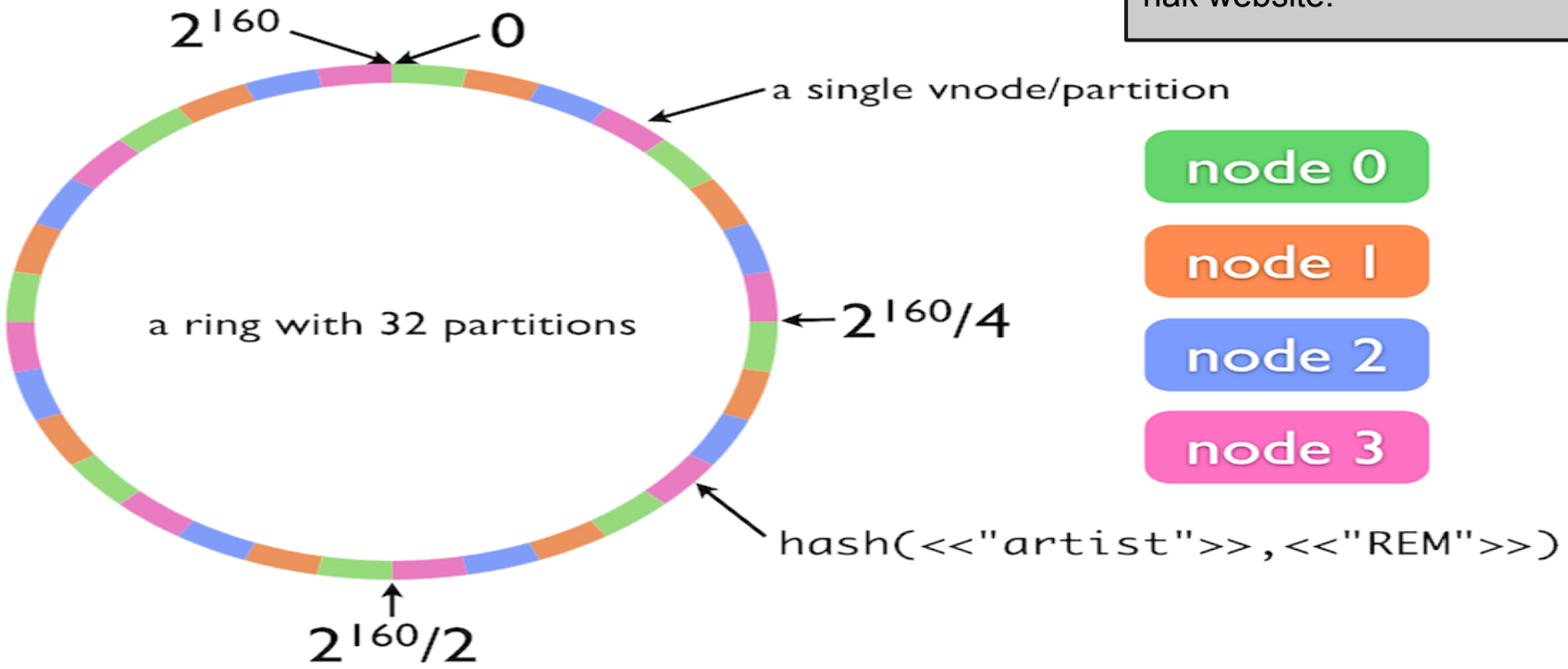
- cd dev
- ls
- dev1 is already started : ps -ef | grep smp
  - `dev2/bin/riak-admin cluster join dev1@127.0.0.1`
  - `repeat for dev2,dev3,dev4 nodes`

Concepts

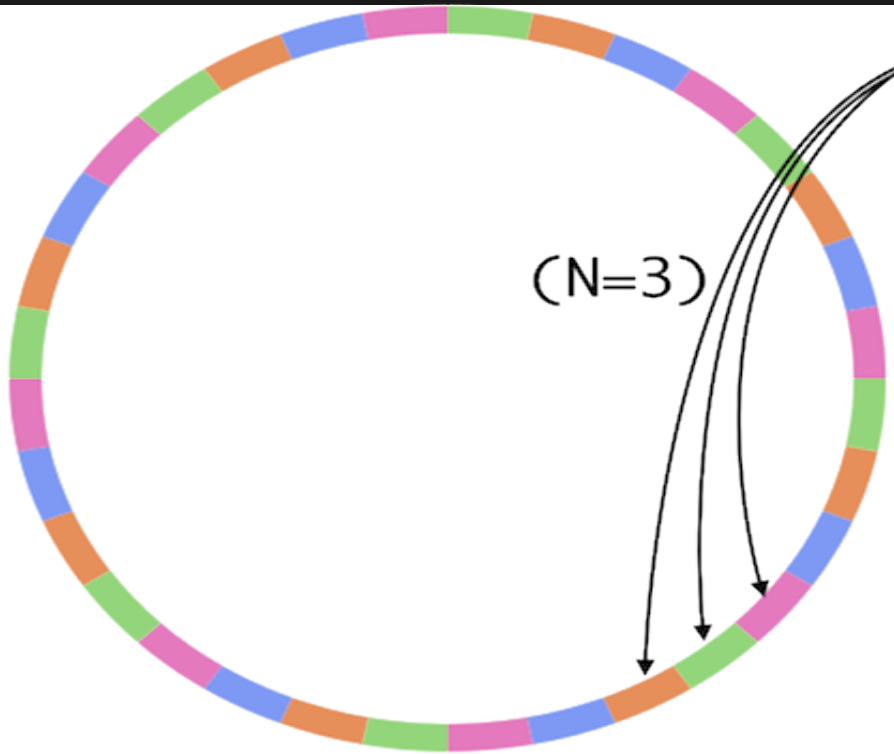
- Ring
- Node
- Partitions
- VNode

Ring

* This is taken from riak docs on riak website.



Replication



Intelligent Replication

* This is taken from riak docs on riak website.

W

- Number of nodes written to before returning success

R

- Number of nodes read from before returning success

Bucket Properties

- `curl -v http://127.0.0.1:10018/buckets/shop-carts/props | python -mjson.tool`

Conflict Resolution

- Last update automatically wins (Implicit)
- All versions are returned to user-User will decide which version to choose

Lets play with cluster

- bring down one node and still it works!
- write even if one node is alive

Link Walking

- Link is an One way relationship
- Link walking is a query option

Search

- distributed and full text search
- Solr-like interface via HTTP
- Exact match queries
    - Wildcards
    - Inclusive/exclusive range queries o AND/OR/NOT support
    - Grouping
    - Prefix matching
    - Proximity searches
    - Term boosting

Case Studies

- Best Buy
- Ideeli
- Amazon dynamo

References

- NoSql Distilled *by Pramod Sadalage and Martin Fowler*
- Little Riak *by Eric Redmond*
- http://docs.basho.com/riak/latest/