Basho & Riak


basho

Who are Basho?

- The creators and developers of Riak & Riak CS
- Founded in 2008 by ex-Akamai staff
- Experts in distributed systems
- Offices across the USA, EMEA & Japan
- Providing Professional Services, Customer Support


riak

What is Riak?

- Riak is a fast, reliable, distributed and highly available key-value store.
- Inspired by Amazon Dynamo white paper.
- Written in Erlang with come C/C++.
- Open Source.
- Apache 2.0 licensed.

What is Riak?

Bucket	Key	Value
	Key	Value
	Key	Value
	Key	Value
Bucket	Key	Value
	Key	Value
	Key	Value
	Key	Value

KV with extras

- Secondary indexing (2i)
- Data expiry (TTL)
- Search
- Map Reduce
- Commit hooks
- HTTP and PB interfaces

riak is the

ops-friendly

database

basho

cluster of **nodes**

DISTRIBUTED

performance through **concurrency**

basho

all **nodes** participate **equally**

MASTERLESS

no **single point** of **failure**

easily add or **remove** nodes

SCALABLE

linear scalability

MULTIPLE PLATFORMS

replicas of stored data

HIGHLY AVAILABLE

redundancy

erlang core

FAULT TOLERANT

self healing

basho

Riak strengths

basho

Riak Strengths

- Fast - KV operations are low latency with minimal disk seeks.
- Scalable - Add nodes to get more CPU/mem/disk/IOPS.
- Concurrent - Erlang OTP gives you concurrency through multiple processes supporting multiple operations.
- Available - Any nodes accepts reads/writes.
- Fault tolerant - Erlang OTP gives you process hierarchy and crash support. The cluster transparently survives node crashes.

Riak is not an RDBMS

basho

Why use Riak?

basho

For Operators

- What's important as an operator?
  - Simplicity
  - Fault Tolerance
  - High-availability
  - Monitoring
  - Excellent support (Community & Enterprise)

Simplicity

- Ease of configuration
- Management & Troubleshooting
- Rolling upgrades
- Provisioning
- Horizontally scalable
- Commodity hardware

Fault Tolerance

- All nodes participate equally - no single point of failure (SPOF)
- All data is replicated
- Cluster transparently survives...
  - Node failure
  - Network partitions
- Built on Erlang/OTP

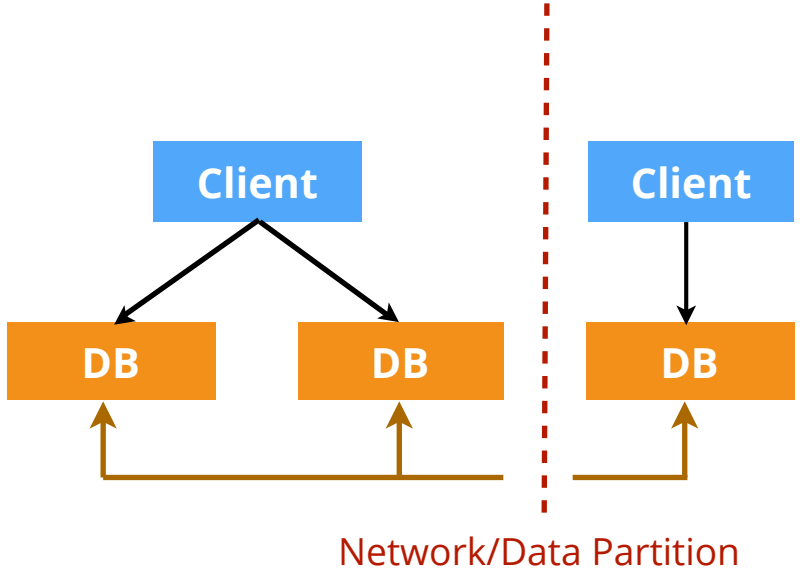
High Availability

- Masterless
- Tunable availability/consistency
- Fallbacks are used when nodes are down
  - Hinted-handoff
  - Ownership-handoff

CAP Theorem

C = Consistency
A = Availability
P = Partition Tolerance

Cap theorem states that a distributed shared data system can at most support 2 out of these 3 properties



Monitoring

- Nagios plugin
- Command line - riak-admin
- HTTP - /stats
- Enterprise
  - JMX
  - SNMP

Support

- Open Source
  - Community
  - Mailing list
  - IRC
  - docs.basho.com - For Operators
- Enterprise
  - Telephone, Email & 24x7x365 On-Call Support

For Developers

- What's important as a developer?
  - Simplicity
  - Supported languages
  - Feature set
  - Performance
  - Excellent support (community & enterprise)

Simplicity

- Simple to spike (Five-minute install)
- No data normalisation
- No need to design for sharding/scaling
- Supported client libraries

Client Libraries

- Client libraries supported by Basho:
  Python, Ruby, Java, Erlang (PB)
- Community supported languages and frameworks:
  C/C++, Clojure, Common Lisp, Dart, Django, Go, Grails, Griffon, Groovy, Erlang, Haskell, .NET, Node.js, OCaml, Perl, PHP, Play, Racket, Scala, Smalltalk

Client Types

- REST based HTTP Interface

  Easy to use from command line (curl) and simple scripts.

- Protocol Buffers

  Optimized binary encoding standard developed by Google. More efficient and faster than HTTP interface.

Features

- Read-repair
- Active Anti Entropy
- Tunable availability/consistency
- Conflict Resolution
- Multiple storage backends with specific features
- Map Reduce
- Replication in Riak EE MDC

Performance

- Near linear performance increase when scaling
- Perfect for high IOPS requirements
- Perfect for heavy write scenarios due to masterless architecture

Support

- Community
- Mailing list
- IRC
- docs.basho.com - For Developers
- Relational to Riak white paper
- Enterprise access to Engineering