

Technology Note

Riak – a distributed, decentralised data storage system

What is Riak?

Riak is a scalable, highly-available, distributed open-source database built around a flexible distributed systems framework. It is written primarily in Erlang.

Key Facts

- Dynamo-based a faithful adaptation of Amazon's Dynamo model
- Cloud-ready elastic architecture means you can grow clusters dynamically without downtime
- Master-less no single point of failure
- Fault tolerant survive outages with no data loss
- Multi-Data Centre write-available, masterless replication
- Linearly scalable adding 10% more nodes means 10% more capacity
- No sharding consistent hashing means 0% downtime

Buckets, Keys, Values

Riak organises data into Buckets, Keys, and Values. Values (or objects) are identifiable by a unique key, and each key/value pair is stored in a bucket.

Buckets are essentially a flat namespace, mainly significant for their ability to allow the same key name to exist in multiple buckets and to provide some per-bucket configurability for things like replication factor and pre/post-commit hooks.

The Riak API

Basho initially implemented both a native Erlang interface, and a HTTP (often called "RESTful") API that allowed users to manipulate data using standard HTTP methods: GET, PUT, POST and DELETE. With the 0.10 release, it added support for accessing Riak using a Protocol Buffers Client interface.

Versioning

Each update to a Riak object is tracked by a vector clock. Vector clocks determine causal ordering and detect conflicts in a distributed system.

Each time a key/value pair is created or updated in Riak, a vector clock is generated to keep track of each version and ensure that the proper value can be determined if there are conflicting updates.

To resolve update conflicts on Riak objects, Riak can either allow the last update to automatically "win", or it can return both versions of the object to the client, letting it resolve the conflict on its own.

Languages

The core Basho Development Team currently supports libraries for Erlang, JavaScript, Java, PHP, Python and Ruby. In addition, there are community contributed projects for .NET, JavaScript, Python (and Twisted), Griffon, Perl, and Scala.

The Riak Cluster

Central to any Riak cluster is a 160-bit integer space (often referred to as "the ring") which is divided into equally-sized partitions.

Physical servers, referred to in the cluster as "nodes", run a certain number of virtual nodes, or "vnodes". Each vnode will claim a partition on the ring. The number of active vnodes is determined by the number of partitions into which the ring has been split, a static number chosen at cluster initialisation.

All nodes in a Riak cluster are equal. Each node is fully capable of serving any client request. This is possible due to the way Riak uses consistent hashing to distribute data around the cluster.

A Riak cluster grows and shrinks dynamically, meaning Riak will automatically re-balance data as nodes join and leave the cluster.

Data Replication

Replication is fundamental and automatic in Riak, providing security that data will still be there if a node in a Riak cluster goes down. All data stored in Riak will be replicated to a number of nodes in the cluster according to the bucket's n_val property.

Querying and Query Languages

Riak relies on MapReduce to perform queries that exceed the limitations that come with the basic key/value storage model. Users can write their MapReduce queries in either Erlang or Javascript.

Benefits

- Low total cost of operations: a lower capital investment and headcount requirement, but with higher performance and reliability.
- Simplicity: it is simple to use and simple to scale
- A large and ever-growing community of users: the open-source community of developers provides additional features and support.

*riak

Technical Data

Developer: Basho Technologies

Operating system: Linux, Mac OS X

License: Apache License 2.0

Written in: Erlang, C, and a small amount of Javascript

Used by

AOL Best Buy Boeing Citigroup Comcast Hova Networks Joyent Linkfluence MIG Mochi Media Opscode Vibrant Wikia Yammer

Sources

wiki.basho.com

Wikipedia

Erlang Solutions 29 London Fruit & Wool Exchange, Brushfield Street London, E1 6EU United Kingdom

Phone +44 (0)20 7456 1020 info@erlang-solutions.com www.erlang-solutions.com