

RIAK KV 2.0 PROVIDES APPLICATION DEVELOPERS ENHANCED USABILITY AND SCALABILITY FOR FASTER

development of highly available, globally scalable applications while ensuring operational simplicity.

As large enterprises and major media companies continue to embrace and deploy distributed NoSQL solutions, resiliency, scalability, and operational simplicity top their list of requirements.

Riak KV is architected for high availability, ease of operations, and massive scalability. It is a distributed NoSQL database optimized for private, public, or hybrid clouds. Riak KV has proven to be operationalized at lower costs than traditional relational databases and is easy to manage at scale. Riak KV integrates with Apache Spark, Redis Caching, Apache Solr, and Apache Mesos to reduce the complexity of integrating and deploying other Big Data technologies.

Riak KV 2.0 includes highly innovative functionality that reduces code complexity and enhances operational simplicity.

RIAK KV 2.0 BENEFITS

ENHANCED SCALABILITY AND USABILITY — Riak Search leverages Apache Solr for enhanced search and query functionality.

REDUCED CODE COMPLEXITY — New data types automatically handle complex conflict resolution without custom client side logic.

ENHANCED OPERATIONAL SIMPLICITY

- Easily control who can run specific commands with integrated authentication and authorization.
- Simplified configuration makes it simple to integrate with automation frameworks.
- Manage bucket properties more effectively with bucket types.
- Optimize access to frequently used data with tiered storage.

NEW FEATURES OF RIAK KV ENTERPRISE 2.0 INCLUDE:

- **Riak Search:** The completely redesigned Riak Search leverages Apache Solr, enabling integration with a wider variety of existing software and commercial solutions through client query APIs.
- **Built-in data types:** In addition to the counter data type, Riak KV now includes sets, registers, flags and maps as new distributed data types to simplify application development, while providing high availability and supporting Riak's partition tolerance characteristics.
- **Security:** Riak KV now provides authorization and authentication to manage users and groups. Security measures can now be applied within Riak KV itself, including data access and functional permissions.
- **Simplified configuration management:** Riak KV's operational simplicity continues to improve by consolidating configuration information and storing it in an easy-to-parse, transparent format.
- **Efficient bucket management:** With bucket types, users can now create and administer bucket properties and apply them to a collection of buckets, improving overall efficiency.
- **Tiered storage:** Riak KV also allows LevelDB users to split data files across two mount points, based on access patterns, to optimize for low latency access of the most frequently accessed data.

RIAK SEARCH



Riak Search brings together the strengths of Riak's horizontally scalable, distributed database with the powerful full-text search functionality of Apache Solr.

This allows for distributed, scalable, transparent indexing and querying of Riak data values.

Riak Search monitors for changes to data in Riak KV and propagates those changes to indexes managed by Solr. Each node in the Riak KV cluster now also supervises an instance of Solr. Each Solr instance houses indexes for data in the Riak KV node.

When queried, Riak Search accepts standard Solr queries and expands them to distributed search queries behind the scenes. Distributed search queries target multiple Solr instances to provide a complete result set across replicas.

WHY RIAK SEARCH OVER SOLR ALONE?

- Riak Search listens for changes in key/value data and makes the appropriate changes to Solr indexes
- Riak Search takes a user query on any node and converts it to a Solr distributed search
- Riak Search takes index creation commands and disseminates that information across the cluster
- Riak Search communicates and monitors the Solr OS process

Riak Search now makes it much easier to ask more complex questions of the data stored in Riak KV. In addition to basic ad-hoc and range queries, Riak Search provides a number of powerful search enhancements, including:

- Support for various MIME types, including JSON, XML, plain text and Riak Data Types
- Analyzer, token and filter support for 30+ languages
- Term boosting
- Sorting
- Pagination
- Scoring and ranking based on result relevancy
- Result snippet highlighting
- Search queries as input for MvmapReduce jobs
- Protocol Buffer interface and Solr interface via HTTP

Riak Search also includes Active Anti-Entropy, which keeps search indexes fresh, over time, as data is manipulated in Riak KV. With Riak Search, Riak KV is responsible for your data and Solr is responsible for your index. Riak KV ensures that your indexes in Solr are up-to-date as changes occur to your data. So, unlike relational databases, Riak KV does all of the work to keep things aligned so that you don't have to.

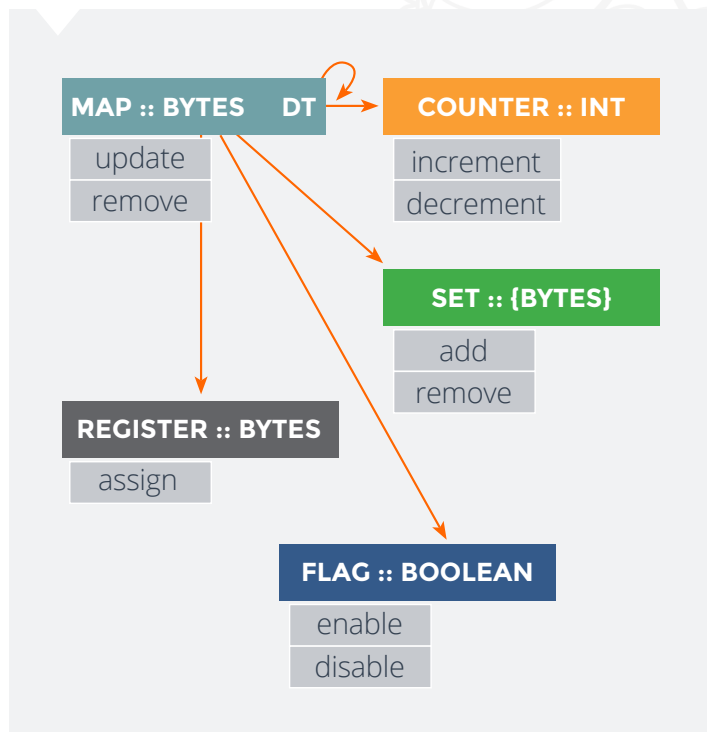
RIAK DATA TYPE

Developer friendly distributed data types help track updates in an eventually consistent environment. Riak data types are pre-built so there is no complex, client-side resolution logic required when using these server-side data types.

In addition to counters, Riak KV 2.0 data types include flags, sets, registers, and maps.

“ The new advanced data types in Riak KV 2.0 are a game changer for us because it makes it simple for us to manage our data model at a scale that supports over a billion devices all around the world. ”

– Weston Jossey, Head of Operations at Tapjoy



DATA TYPES	USE CASES	CONFLICT RESOLUTION RULE
Counters (v1.4) – keep track of increments/decrements	<ul style="list-style-type: none"> Track number of page “likes” or number of followers 	<ul style="list-style-type: none"> Each actor keeps an independent count for increments and decrements Upon merge, the pairwise maximum of any two actors will win (e.g. if one actor holds 172 and the other holds 173, 173 will win upon merge)
Flags – enabled/disabled	<ul style="list-style-type: none"> Has a tweet been re-tweeted Is a user eligible for preferred pricing 	enable wins over disable
Sets – collection of binary values	<ul style="list-style-type: none"> List items in an online shopping cart UUIDs of a user’s friends in a social networking app 	If an element is concurrently added and removed, the add will win
Registers – named binary with values also binary	<ul style="list-style-type: none"> Store user profile names Store primary search location for a search engine user 	The most chronologically recent value wins, based on timestamps
Maps – supports nesting of multiple data types	<ul style="list-style-type: none"> Store user profile data composed <ul style="list-style-type: none"> - register user_name - flag email_notifications - counter site_visits 	If a field is concurrently added or updated and removed, the add/update will win

SECURITY

Riak KV 2.0 introduces authentication and authorization. Riak KV administrators can grant permissions to control access to objects, change bucket properties and run MapReduce jobs.

Authentication: PAM functionality supports integration with other authentication platforms, such as LDAP. Even though groups of users may be granted functional permissions, authentication must be at the individual level. Sources are used to define authentication mechanisms and a user cannot be authenticated to Riak until a source is defined.

Authorization: Users can be assigned to one or more groups for convenience in managing permissions. Permissions are set at the bucket type level or as a global setting for all buckets. Creating, modifying, and deleting user characteristics and granting users selective access to Riak KV functionality, controls authorization. Users can be assigned one or more of the following characteristics: username, groups, password.

SIMPLIFIED CONFIGURATION MANAGEMENT

Riak KV configuration settings now have an easy-to-parse, transparent format enabling simpler integration with automation frameworks. Configuration settings are now stored in riak.conf. The riak.conf file is used to set a wide variety of attributes for the node, including the storage backend that the

node will use to store data, the location of SSL-related files, sibling resolution parameters, and more.

BUCKET TYPES

Riak KV bucket properties can now be managed using bucket types. Bucket types enable you to create bucket configurations and assign those configurations to as many buckets as needed. Buckets associated with a bucket type only have their properties changed when the type is changed. The riak-admin bucket-type interface enables you to manage bucket configurations on the operations side without client side coding.

TIERED STORAGE

One of the backends for Riak KV is LevelDB. As part of standard operation, LevelDB automatically identifies frequently accessed information and places it in a higher "level" in its file structure. Using these levels, Riak now provides the ability to split data files across mount points with different I/O performance. You configure when to cut over data from the faster to the slower mount, and LevelDB does the rest ensuring predictable latency at the 99th percentile and above.



ABOUT BASHO TECHNOLOGIES

Basho, the creator of the world's most resilient databases, is dedicated to developing disruptive technology that simplifies enterprises' most critical distributed systems data management challenges. Basho has attracted one of the most talented groups of engineers and technical experts ever assembled devoted exclusively to solving some of the most complex distributed systems challenges presented by Big Data and IoT.

Basho's database, Riak® KV, the industry leading distributed NoSQL database, is used by fast growing Web businesses and by one-third of the Fortune 50 to power their critical Web, mobile and social applications. Built on the same foundation, Basho introduced Riak TS, which is the first enterprise-ready NoSQL database specifically optimized to store, query and analyze time series data. Basho also provides Riak integrations for a variety of Big Data technologies like Apache Spark, Redis, Mesos, and Apache Solr.

For more information visit Basho.com which is full of interesting use cases, customer case studies and product detail, or docs.basho.com for technical documentation.