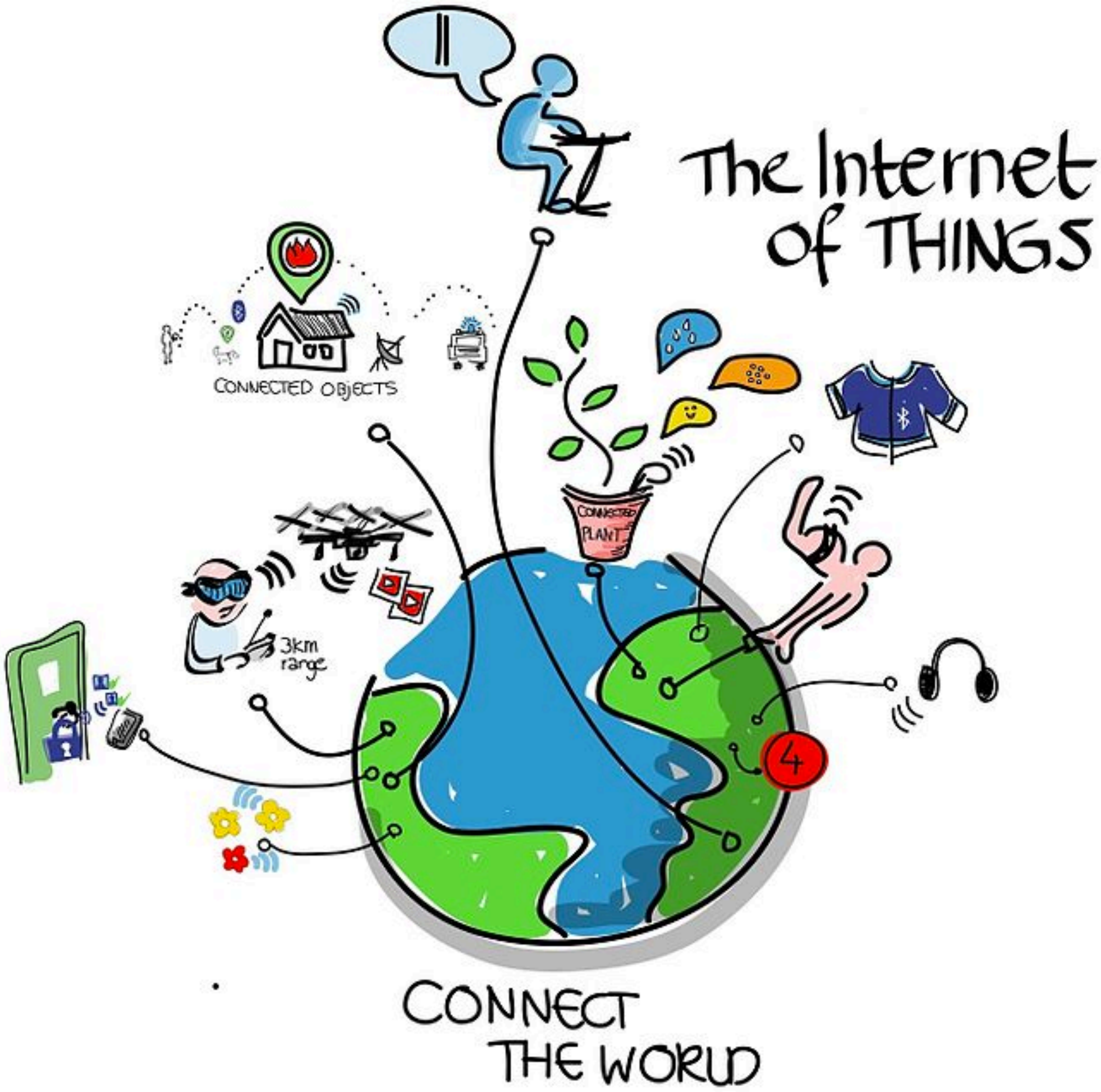
#IoT #BigData

Seema Jethani @seemaj @basho

Why should we care?



Source: http://en.wikipedia.org/wiki/Internet_of_Things

Motivation for Specialized Big Data Systems

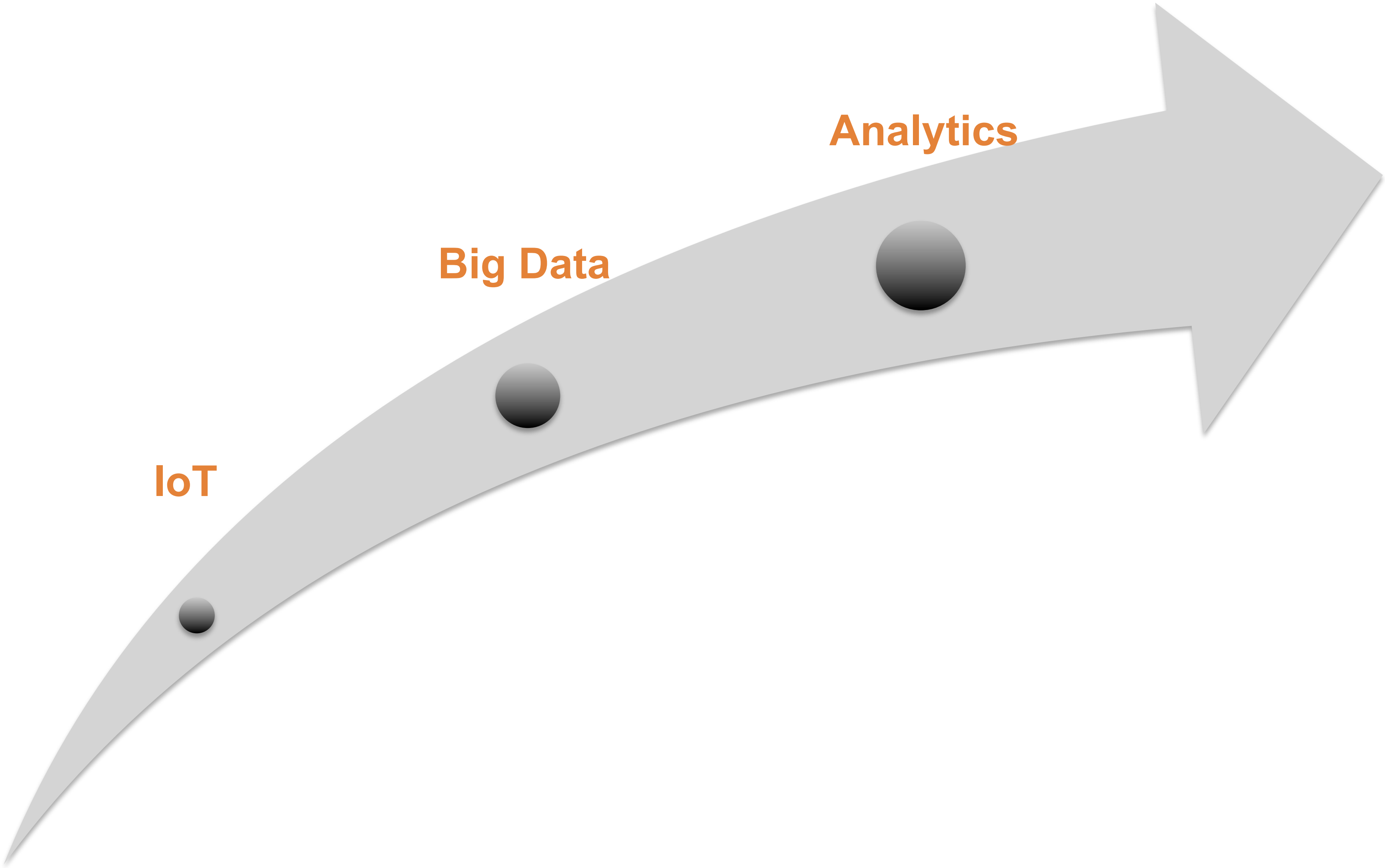
Rate of data capture started soaring

Traditional data warehouses and RDBMS systems could not keep up

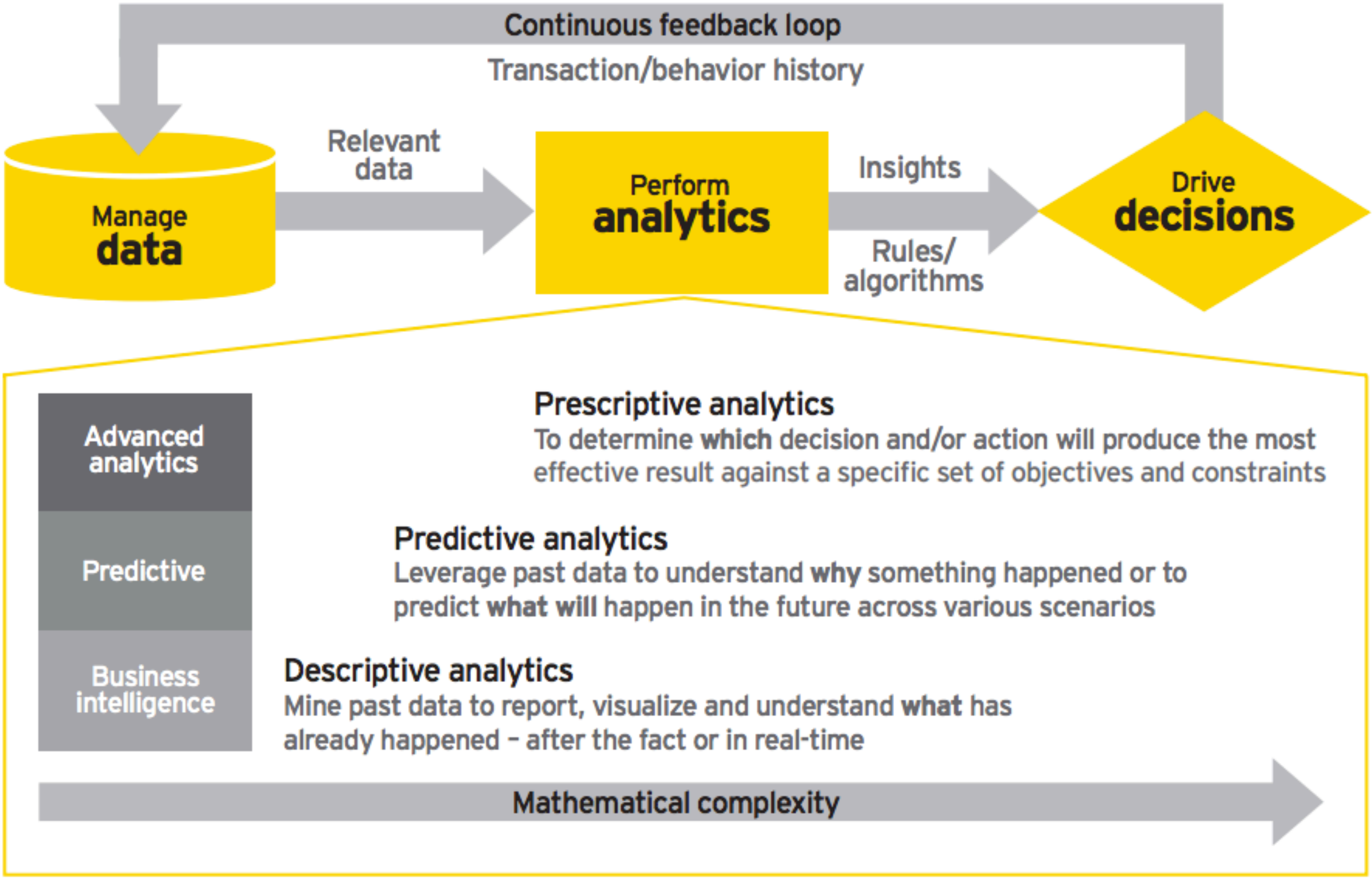
Specialized Big Data systems were introduced -

Distributed
Cluster-based
Commodity priced
Linearly scalable
Process parallelly
Node redundant

Where is the value?



The Big Data Value Chain



Continuous feedback loop

Transaction/behavior history

Manage **data**

Relevant data

Perform **analytics**

Insights

Rules/ algorithms

Drive **decisions**

Advanced analytics

Predictive

Business intelligence

Prescriptive analytics
To determine **which** decision and/or action will produce the most effective result against a specific set of objectives and constraints

Predictive analytics
Leverage past data to understand **why** something happened or to predict **what will** happen in the future across various scenarios

Descriptive analytics
Mine past data to report, visualize and understand **what** has already happened – after the fact or in real-time

Mathematical complexity

Source: EY

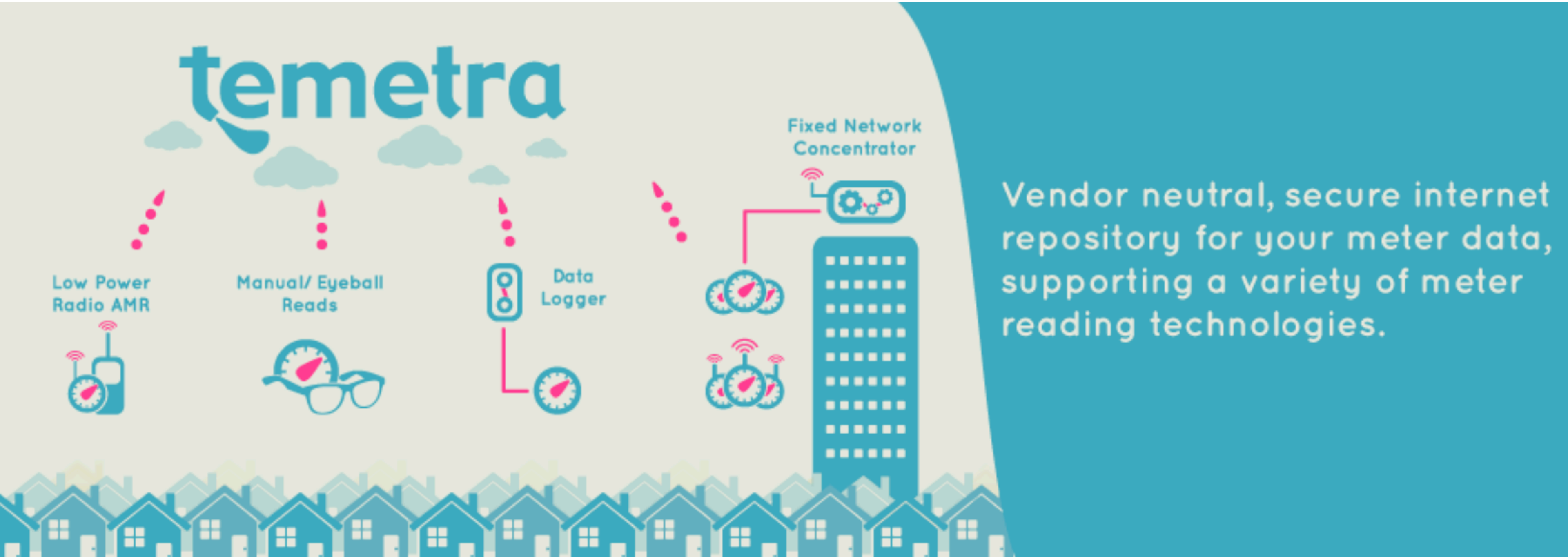
A tale of meters

A leading provider of cloud-based meter data management for water, gas and heat meters.

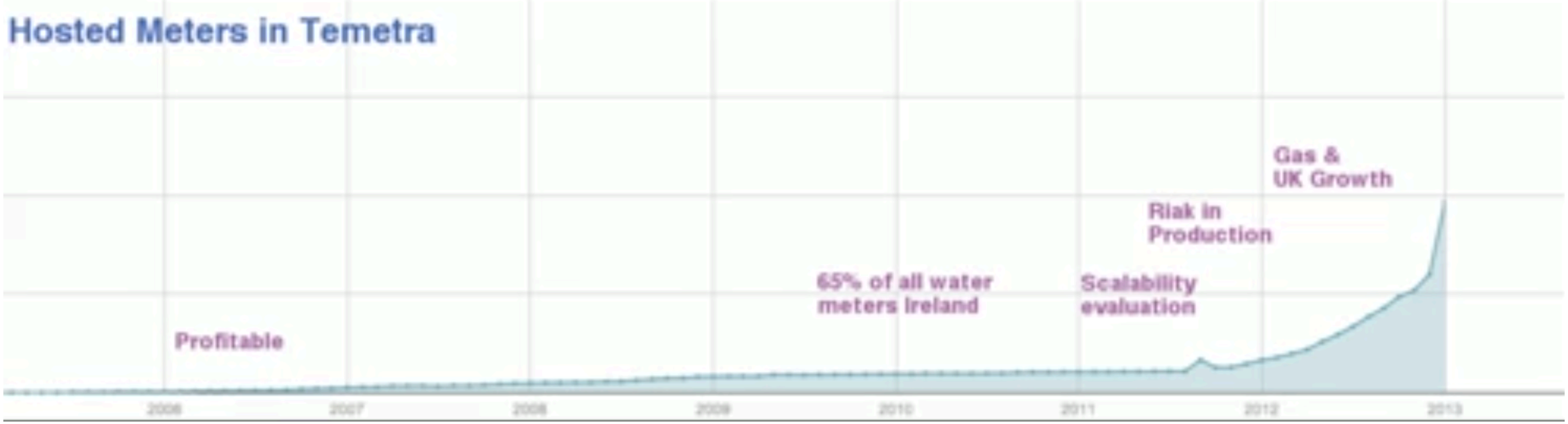
Data is revenue critical

Data loss is non acceptable

Reliability and availability trumps all


temetra
Fixed Network Concentrator
Low Power Radio AMR
Manual/ Eyeball Reads
Data Logger
Vendor neutral, secure internet repository for your meter data, supporting a variety of meter reading technologies.

Temetra growth



Shape of Temetra data

Big data: Millions of meters generating billions of data points

Meters in 2000: four data points a year

Meters in 2013: up to 35,000 data points a year

Enormously high data ingress with relatively few reads

Small number of users (1000s, 100s logged in)

Slow moving, ever increasing data

– Audit trails , photos

Traditional databases were no longer suitable

– Selected Riak to help manage data growth

Data Types in Riak

Developer friendly distributed data types to help track updates in an eventually consistent environment

Pre-built data types - no complex, client-side resolution logic is required when using server-side data types

First introduced in 1.4 as counters, 2.0 adds:

SETS

FLAGS

REGISTERS

MAPS

CRDTs

CRDTs expose simple, well-known data types but with an internal structure that makes it safe to update them without any coordination between writers and without any loss of information in the face of concurrency.

The “C” in CRDT can stand for three different things

- “convergent” datatypes ensure that disparate states converge to a single value
- “commutative” datatypes are updated with commutative operations
- “conflict-free” datatypes if you wish to describe both/either at once without referring to the specifics of your internal choices.

Benefits of CRDT

- We don’t need to send duplicate data.

- For CmRDTs It doesn’t matter what order the two requests happen in, the outcome will be the same.

- There is no possibility of a datatype returning siblings, making client code that much easier.
  - Conflicting values are known as siblings
  - Siblings arise in a couple cases.
    - 1. A client writes a value using a stale (or missing) vector clock.
    - 2. Two clients write at the same time with the same vector clock value.

Data Types in Riak

DATA TYPE	USE CASE
Counters (v1.4) – keep track of increments/decrements	- Track number of page "likes" or number of followers
Flags – enabled/disabled	- Has a tweet been re-tweeted - Is a user is eligible for preferred pricing
Sets – collection of binary values	- List items in an online shopping cart - UUIDs of a user's friends in a social networking app
Registers – named binary with values also binary	- Store user profile names - Store primary search location for a search engine user
Maps – supports nesting of multiple data types	- Store user profile data composed register user_name flag email_notifications counter site_visits

Conflict Resolution of data types in Riak

DATA TYPE	USE CASE
Counters (v1.4) – keep track of increments/decrements	– 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	– enable wins over disable
Sets – collection of binary values	– If an element is concurrently added and removed, the add will win
Registers – named binary with values also binary	– The most chronologically recent value wins, based on timestamps
Maps – supports nesting of multiple data types	– If a field is concurrently added or updated and removed, the add/update will win

A CRDT example

- Assume that the bucket type map is of a map datatype This command will insert a map object with two fields (name_register and pets_set).

-
  ```
  curl -XPOST "$RIAK/types/map/buckets/people/keys/joe” -H "Content-Type:application/json”
  -d ’{
  "update": {
  "name_register": "Joe »
  "pets_set": {
  "add_all": "cat »
  }
  }
  }’
  ```
- Next, we want to update the pets_set contained within joe ’s map. Rather than set Joe’s name and his pet cat, we only need to inform the object of the change. Namely, that we want to add a fish to his pets_set .

-
  ```
  curl -XPOST "$RIAK/types/map/buckets/people/keys/joe” -H "Content-Type:application/json”
  -d ’{
  "update": {
  "pets_set": {
  "add": "fish”
  }
  }
  }'
  ```

Querying and analyzing the data

e.g. Find the closest post code to a particular post code

Riak Search combines the operational simplicity and fault tolerance of Riak with the powerful search functionality of Apache Solr

Allows for distributed, scalable, transparent indexing and querying of Riak data values

Combine CRDTs, Search with pre and post processing of data to analyze data in real time

Riak search queries

Query Parameters

- Exact match
- Globs
- Inclusive/exclusive range queries
- AND/OR/NOT
- Prefix matching
- Proximity searches
- Term boosting
- Sorting
- Pagination

Features:

- Scoring and ranking for most relevant results
- Search queries as input for MapReduce jobs
- Active Anti-Entropy for automatic index repair
- Multiple languages, geo-spatial search, tokenizers and filters
- Supports various MIME types (JSON, XML, plain text, data types) for automatic data extraction

Write it like Riak Query it like SOLR

Every node in a Riak cluster has a corresponding operating system (OS) process running a JVM which hosts Solr on the Jetty application server.

Riak Search listens for changes in key/value (KV) 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

Example Search using SOLR

Indexes may be associated with zero or more buckets. At creation time, however, each index has no associated buckets

To associate a bucket with an index, the bucket property yz_index must be set to the name of the index you wish to associate. Conversely, in order to disassociate a bucket you use the sentinel value _dont_index_.

Many buckets can be associated with the same index. A bucket *cannot* be associated with many indexes—the yz_index property must be a single name, not a list

SOLR example

Schemas explain to Solr how to index fields
Indexes are named Solr indexes against which you will query
Bucket-index association signals to Riak *when* to index values

Search Index with default schema:

```
curl -XPUT $RIAK_HOST/search/index/famous \ -H 'Content-Type: application/json' \ -d
'{"schema":"_yz_default"}’
```

Bucket Index association:

```
riak-admin bucket-type create animals '{"props":{"search_index":"famous"}}' riak-admin bucket-type activate
animals
```

Write data:

```
curl -XPUT "$RIAK_HOST/types/animals/buckets/cats/keys/liono" \ -H'content-type:application/json' \ -
d'{"name_s":"Lion-o", "age_i":30, "leader_b":true}' curl -XPUT "$RIAK_HOST/types/animals/buckets/cats/
keys/cheetara" \ -H'content-type:application/json' \ -d'{"name_s":"Cheetara", "age_i":28, "leader_b":false}’
```

Query:

```
curl $RIAK_HOST/search/query/famous?wt=json&q=name_s:Lion* | jsonpp
```

```
{ "numFound": 1, "start": 0, "maxScore": 1.0, "docs": [ { "leader_b": true, "age_i": 30, "name_s": "Lion-o",
"_yz_id": "default_cats_liono_37", "_yz_rk": "liono", "_yz_rt": "default", "_yz_rb": "cats" } ] }
```

Summary

- IoT deployments will generate large quantities of data that need to be processed and analyzed
  “IoT will mean really, really Big Data” (InfoWorld)

- We need to design for analytics – “creating a strategy that sees data more as a supply chain than a warehouse” Mike Redding, Accenture

- Not all data is made equal – we need to find the important and act on it

- Data driven decision making will be key in achieving business success

Questions

Interested in Tech Talk? smoder@basho.com