Putting together the platform: Riak, Redis, Solr and Spark

Bryan Hunt
$ whoami

Bryan Hunt

Client Services Engineer

@binarytemple
Minimum viable product - the ideologically correct doctrine

1. Start with what you have
2. Solve problems as they arise
3. Get to the **point where you can’t scale**
4. Use a **more scalable** solution
SAVE $100 ON SELECT ASUS 2-IN-1 LAPTOPS
Powered by Intel®. Plus, save an additional $50-$150 with exclusive College Student Deals.

Shop Laptops in the Student Tech Center

Free upgrade to Windows 10.

Learn more about Windows 10

EXPERT SERVICE. UNBEATABLE PRICE.
Small ecommerce site

- Sessions stored in browser cookie
- As the cart is mutated it is written to the database
- Abandoned carts timeout and items are returned to inventory
“We’re getting slow”
that means we’ve succeeded
Billions of Data Points

Millions of Users, 10 Million Meters in EU

Only 4 people devops team

2000
4 DATA POINTS PER YEAR
(traditional gas meter)

2013
35,000+ DATA POINTS PER YEAR
(digital meter)
More users == slow read performance

```python
if !cache $item
    db.fetch($item)
```

- Latency increases when not cached
- Caching layer added with above logic
- Need for cache invalidation logic
- Cache warming....
- Oh my!

![Diagram showing Web Client and RDBMS with Cache in between]
Web store - now support mobile clients

- Mobile client added
- RDBMS write locks increase
- Latency still increases
- Caching layer added capacity but invalidation is a pain
- Write locks are still an issue
More and more demands… make it searchable…

How can it take 3 weeks to add a simple feature?
Web store - growing pains

- SQL profiling/optimization
- Search is separated
- Don’t forget to populate the index
- Never, never run *that* query
Product Catalogue – The Feature Backlog Continues

- Scale and Availability
- Performance
- Analytics
- Mobile
- Search
Enter the Riak
Scalable, Available, Fault Tolerant

Riak has a masterless architecture in which every node in a cluster is capable of serving read and write requests.
Write Availability and Consistency

Are they mutually exclusive?
Riak Data Types (Convergent Replicated Data Types or CRDTs) are a developer-friendly way to keep track of updates in an eventually consistent environment:

- **Counter**
  Keeps tracks of increments and decrements on an integer

- **Flag**
  Values limited to *enable* or *disable*

- **Set**
  A collection of unique binary values that supports add and remove operations on one or more values

- **Map**
  Supports the nesting of Riak Data Types

- **Register**
  A named binary field that can only be used as part of a Map

Conflicts?
Multi-cluster Replication
Dealing with Search at scale

Distributed Full-Text Search
Standard full-text Solr queries automatically expand into distributed search queries for a complete result set across instances.

Index Synchronization
Data is automatically synchronized between Riak KV and Solr using intelligent monitoring to detect changes, and propagates those to Solr indexes.

Auto-Restart
Monitor Solr OS processes continuously and automatically start or restart them whenever failures are detected.
Make Search Better
we need to get a competitive advantage
Web store: One step closer to success

Legacy Infrastructure

Product Catalogue Relational DB

Middleware
Python/Ruby/Node

JSON

Riak KV
Store JSON
Index & Search via integrated Solr

Web / Mobile

Cache Solr Results

Product Catalogue Pipeline
Basho Data Platform
What it provides..

• Easy to use caching layer
• Automatically indexed search
• Scalable, highly available persistence layer
• Simple pipelining to data analysis
Caching
Redis - why ?
Why Use Redis With The Data Platform?

• Redis, by itself, is approximately 6x faster than Riak for reads

• Redis is not built to scale

With the Basho Data Platform:

• Approximately 5x faster read performance compared to Riak for buckets at an 80% cache hit ratio.
Basho Data Proxy & Redis

Data Platform Basho Proxy - Redis

Start 5 redis nodes with the Service Manager. Each redis instance will bind to its node’s IP address and specified (or default) port.

Start a Basho Proxy node, passing in 5 redis-node IP addresses/ports. The Basho Proxy is now the proxy for all 5 redis nodes.

Redis client makes data request from the Basho Proxy. The proxy in turn looks for the data in redis. If not found, it looks for the data in Riak. If found, it will then write the data to redis and return the data to the requesting client.
Resilience

5 redis nodes running:
- Service Manager
- Redis
  Node 1
  Node 2
  Node 3
  Node 4
  Node 5

Node 5 goes down, still 4 redis nodes running:
- Service Manager
- Redis
  Node 1
  Node 2
  Node 3
  Node 4
  Node 5

Node 1 goes down, node 5 returns to service, 4 redis nodes running:
- Service Manager
- Redis
  Node 1
  Node 2
  Node 3
  Node 4
  Node 5
Basho Data Proxy & Redis

**Redis**

- v2.8.21
  - Redis Cluster (3.0+) need not apply

- Cache Proxy using only GET and SET
  - SET uses a TTL, configured as CACHE_TTL

- Client may use other commands such as:
  - PEXPIRE to invalidate a cached value
  - INFO to integrate with monitoring solutions
  - MONITOR to watch traffic to Redis hosts

**Basho Data Proxy**

- Written in C
- Read-through cache in BDP 1.0
- Write-through cache in a future BDP release
- EE only, until write-through cache is complete and released
- Pre-sharding & connection aggregation built-in
Analytics
Connecting to BDP with Spark

Spark Client → Spark Connector → riakKV
Data Platform Spark Cluster
Spark & Spark Cluster Manager

Spark

- V1.4.1
- Dependencies
  - Java 8
  - Scala 2.10.5

Spark Cluster Manager

- Shipped as a JAR in the BDP extras package
- Enables running a HA Spark cluster
  - Spark Master
  - Spark Stand-by Master(s)
- Implemented as a custom Spark recovery mode
- Stores its metadata in a strongly consistent Riak bucket
Why Basho Data Platform?
SAVE $100 ON SELECT ASUS 2-IN-1 LAPTOPS
Powered by Intel®. Plus, save an additional $50-$150 with exclusive College Student Deals.

Shop Laptops in the Student Tech Center

Free upgrade to Windows 10.
Learn more about Windows 10 ›
Product Catalogue with Basho Data Platform

- Mobile Client
- Web Client
- Cache Proxy
- Product Catalogue Pipeline
- Analytics Services

Diagram showing the integration of Basho Data Platform components: Cache Proxy, Web Client, Mobile Client, Product Catalogue Pipeline, and Analytics Services.
Faster Analysis

More IOPS/TBs

Faster Reads
QUESTIONS?
**Spend Time getting to know us**

<table>
<thead>
<tr>
<th>OPEN SOURCE</th>
<th>ENTERPRISE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basho Data Platform</strong> <em>(code)</em></td>
<td></td>
</tr>
<tr>
<td>• Riak KV with parallel extract</td>
<td></td>
</tr>
<tr>
<td><strong>Basho Data Platform Add-on’s</strong> <em>(code)</em></td>
<td></td>
</tr>
<tr>
<td>• Spark + Spark Connector</td>
<td></td>
</tr>
<tr>
<td><strong>Download a build</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Basho Data Platform, Enterprise</strong></td>
<td></td>
</tr>
<tr>
<td>• Riak EE with multi-cluster replication</td>
<td></td>
</tr>
<tr>
<td>• Spark Leader Election Service</td>
<td></td>
</tr>
<tr>
<td><strong>Basho Data Platform Add-on’s</strong></td>
<td></td>
</tr>
<tr>
<td>• Redis + Cache Proxy</td>
<td></td>
</tr>
<tr>
<td>• Spark Workers + Spark Master</td>
<td></td>
</tr>
<tr>
<td><strong>Contact us</strong> to get started</td>
<td></td>
</tr>
</tbody>
</table>

@basho
@binarytemple
Please remember to rate this session.

Thank you!
Thanks !