Fear & Uncertainty of Open Source

Wes Chow, CTO @ Chartbeat, @weschow

Slides: bit.ly/woeful-achilles
An Average Day in Numbers

- 275,000 beacon requests / sec
- 1M+ messages / sec through Kafka
- 700 EC2 instances
- 10,000 API calls / sec
- 1 PB data warehouse
Databases

- RDS (MySQL, Postgres)
- Citus
- Mongo
- Riak
- HDFS/HBase
- DynamoDB
- Redshift
- Drill
- Kafka
- Memoryfly (proprietary)
“best tool for the job” != “best tool for the company”
Jason: I would like an illustration of a python choking on a large ruby. Also the python is wearing boxer shorts.

me: Anything else?

Jason: I wouldn't mind if the ruby was singing...
## Time Series

<table>
<thead>
<tr>
<th>DATE</th>
<th>TIME</th>
<th>DOMAIN</th>
<th>PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010-01-01</td>
<td>09:00</td>
<td>chartbeat.com</td>
<td>/</td>
</tr>
<tr>
<td>2010-01-01</td>
<td>09:00</td>
<td>chartbeat.com</td>
<td>/some-article-1</td>
</tr>
<tr>
<td>2010-01-01</td>
<td>09:01</td>
<td>chartbeat.com</td>
<td>/some-article-2</td>
</tr>
<tr>
<td>2010-01-01</td>
<td>09:02</td>
<td>chartbeat.com</td>
<td>/some-article-1</td>
</tr>
<tr>
<td>2010-01-01</td>
<td>09:59</td>
<td>chartbeat.com</td>
<td>/</td>
</tr>
<tr>
<td>2010-01-01</td>
<td>09:59</td>
<td>chartbeat.com</td>
<td>/some-article-52</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Home Page Views (SQL)

SELECT
    count(1) AS pv
FROM
    events
WHERE
    time >= '2010-01-01' AND
    time < '2010-01-02' AND
    domain='chartbeat.com' AND
    path = '/'
;
Home Page Views (Actual)

db['events'].find(
    {
        time: {
            $ge: new Date('2010-01-01 09:00:00'),
            $lt: new Date('2010-01-01 10:00:00')
        },
        domain: 'chartbeat.com',
        path: '/'
    }
).count()
Tech Decision Framework

1. Does it solve the problem?
2. How similar is this to something we already use?
4. What is the development cost of using it? Is it hard to integrate? Is it hard to maintain? Do there exist best practices or are we on our own? How far in the future will it remain useful, and is that commensurate with the effort required to deploy?
5. What advantages does it bestow? Is becoming experts an advantage because of differentiated tech, hiring, or filling a community vacuum? Will we gain new capabilities?
6. Does it really solve the problem? Do we understand how it works? Does it make sense from first principles? Do the authors use it?
Nederland probeert klimaat te redden van Trump

Staatssecretaris Sharon Dijskma (Milieu, PvdA) organiseert in de VS een conferentie met de titel ‘Climate First’ - een knipoog naar Trumps campagneslogan ‘America First’.
HUD Database Contenders

- RDS (MySQL)
- Mongo 2.x
- Memcache
- Redis
- HBase
- Cassandra
- Riak (in use with Engaged Headline Testing [HyperLogLog])
Tech Decision Framework

1. Does it solve the problem?
2. How similar is this to something we already use?
4. What is the development cost of using it? Is it hard to integrate? Is it hard to maintain? Do there exist best practices or are we on our own? How far in the future will it remain useful, and is that commensurate with the effort required to deploy?
5. What advantages does it bestow? Is becoming experts an advantage because of differentiated tech, hiring, or filling a community vacuum? Will we gain new capabilities?
6. Does it really solve the problem? Do we understand how it works? Does it make sense from first principles? Do the authors use it?
HUD So Far...

- Scaled up, then scaled down
- Operational edge cases
- CRDTs
- Erlang, source code not useful
- (Riak support was great)
- DynamoDB...
Tech Decision Framework (DynamoDB)

1. Does it solve the problem?
2. How similar is this to something we already use?
4. What is the development cost of using it? Is it hard to integrate? Is it hard to maintain? Do there exist best practices or are we on our own? How far in the future will it remain useful, and is that commensurate with the effort required to deploy?
5. What advantages does it bestow? Is becoming experts an advantage because of differentiated tech, hiring, or filling a community vacuum? Will we gain new capabilities?
6. Does it really solve the problem? Do we understand how it works? Does it make sense from first principles? Do the authors use it?
Riak TS

- Riak TS = Riak KV + structure
- Each K mapped to subset of nodes
- K contains quanta, ex chartbeat.com:20170101:10
- V = LevelDB with subkeys
- K placed randomly
Riak TS Reads

... chartbeat.com:20170101:00
... chartbeat.com:20170101:01
... chartbeat.com:20170101:02
... chartbeat.com:20170101:03
... chartbeat.com:20170101:04
...
Political Capital

The ability to introduce a major technology change.
First Principles

The process of taking what we know about the physical limitations, and backing out characteristics of the system.
Time Series Data Locality

...
20170101:00:foo.com  ->  121
20170101:00:bar.com  ->  2753
20170101:00:chartbeat.com  ->  2
...

(1 hour later)

...
20170101:01:foo.com  ->  189
20170101:01:bar.com  ->  3217
20170101:01:chartbeat.com  ->  1
...
...
Mongo on Disk (simplified and not totally correct)

ctr.0

20170101:00:foo.com 121
20170101:00:bar.com 2753
20170101:00:chartbeat.com 2
...
20170101:01:foo.com 189

20170101:01:bar.com 3217
20170101:01:chartbeat.com 1
...

ctr.1

20170101:05:foo.com 134
20170101:05:bar.com 2543
20170101:05:chartbeat.com 2
...

...
Mongo on Disk (simplified and not totally correct)

ctr.0

<table>
<thead>
<tr>
<th>Time</th>
<th>Host</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>20170101:00:00:foo.com</td>
<td>121</td>
<td></td>
</tr>
<tr>
<td>20170101:00:bar.com</td>
<td>2753</td>
<td></td>
</tr>
<tr>
<td>20170101:00:chartbeat.com</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20170101:01:foo.com</td>
<td>189</td>
<td></td>
</tr>
</tbody>
</table>

ctr.1

<table>
<thead>
<tr>
<th>Time</th>
<th>Host</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>20170101:05:foo.com</td>
<td>134</td>
<td></td>
</tr>
<tr>
<td>20170101:05:bar.com</td>
<td>2543</td>
<td></td>
</tr>
<tr>
<td>20170101:05:chartbeat.com</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Mongo on Disk (collection oriented, still not totally correct)

ctr.0

<table>
<thead>
<tr>
<th>Date</th>
<th>Host</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>20170101:00</td>
<td>foo.com</td>
<td>121</td>
</tr>
<tr>
<td>20170101:01</td>
<td>foo.com</td>
<td>189</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>20170101:05</td>
<td>foo.com</td>
<td>134</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Host</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>20170101:00</td>
<td>bar.com</td>
<td>2753</td>
</tr>
<tr>
<td>20170101:01</td>
<td>bar.com</td>
<td>3217</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>20170101:05</td>
<td>bar.com</td>
<td>2543</td>
</tr>
</tbody>
</table>

ctr.1

<table>
<thead>
<tr>
<th>Date</th>
<th>Host</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>20170101:00</td>
<td>chartbeat.com</td>
<td>2</td>
</tr>
<tr>
<td>20170101:01</td>
<td>chartbeat.com</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>20170101:05</td>
<td>chartbeat.com</td>
<td>2</td>
</tr>
</tbody>
</table>

...
Why is Open Source Important?

- Shared cost of development
- Free as in freedom
- Moral imperative
- Community eats a variety of dogfood
- Wild wild west of ideas
Open vs. Closed Development

Open -- Majority of contributions are by users

Closed -- Majority of contributions by one or a few large stakeholders
Why is Open Source Important?

- Shared cost of development
- Free license, freedom from lockin
- Moral imperative
- Community eats a variety of dogfood
- Wild wild west of ideas
This is Git. It tracks collaborative work on projects through a beautiful distributed graph theory tree model.

Cool. How do we use it?

No idea. Just memorize these shell commands and type them to sync up. If you get errors, save your work elsewhere, delete the project, and download a fresh copy.
The Elasticsearch Saga

- Elasticsearch for log data
- Operationally time consuming
- Amazon Elasticsearch Service (ESS)
- Amazon ESS incompatible: signed requests & no admin endpoints
obviously, a reminder, Elastic official cloud offering, https://www.elastic.co/cloud, doesn't suffer from this limitation of AWS and curator works out of the box with it. This is being helped with the fact that both the devs behind ES and @untergeek work for Elastic and can make sure this happens. Obviously any fork of curator is not officially supported by Elastic.
Interesting. I wonder what AWS included in the metadata. They apparently included the `/_cluster/state` endpoint, which is necessary for Curator, but may not have included all of the necessary information in that endpoint. It does not appear that the individual index settings (which contain the metadata) are there, at least not under the `settings` key.
Additional Reading

- Nadia Eghbal, “How I Stumbled Upon the Internet’s Biggest Blind Spot”
- Tim O’Reilly, “The Architecture of Participation”
- Pieter Hintjens / ZeroMQ, “Collective Code Construction Contract (C4)”