Wikimedia Content API:

A Cassandra Use-case

Eric Evans <eevans@wikimedia.org>
@jericevans

Strangeloop | September 17, 2016


WIKIMEDIA

Our Vision:

A world in which every single human can freely share in the sum of all knowledge.

About:

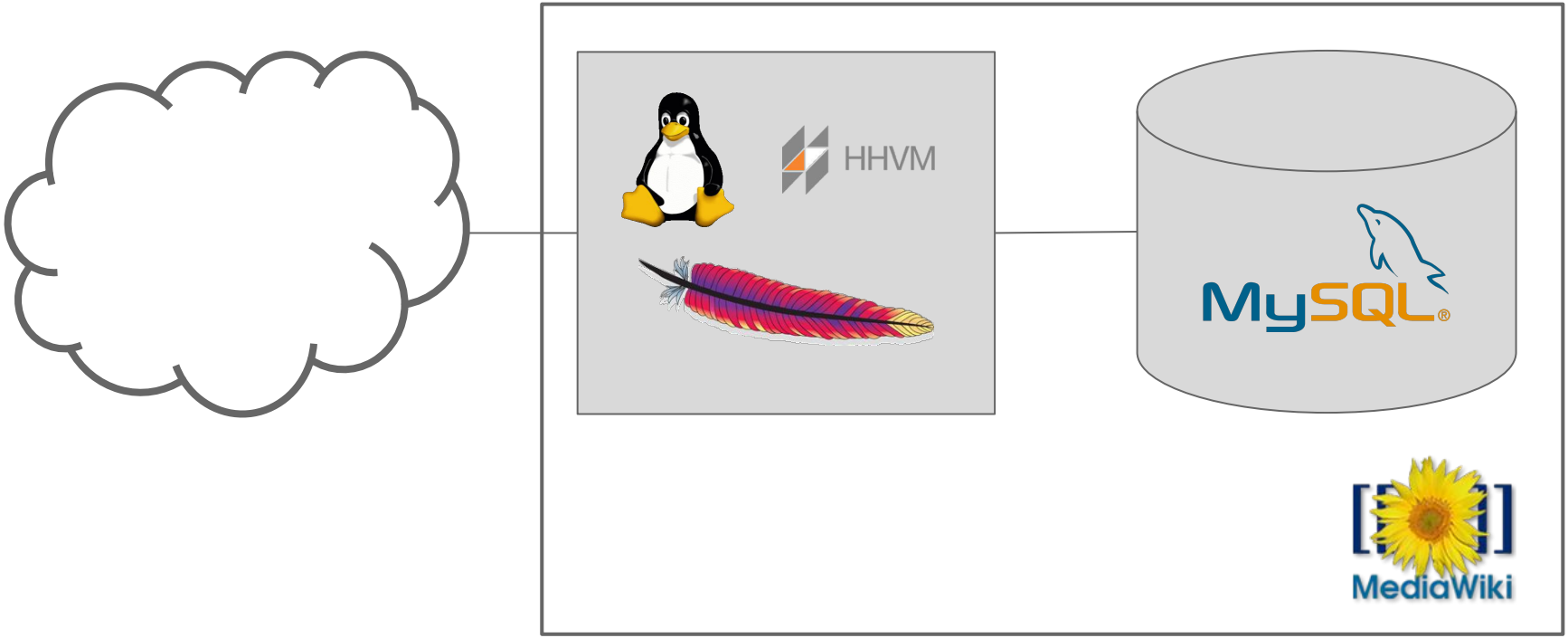
- Global movement
- Largest collection of free, collaborative knowledge in human history
- 16 projects
- 16.5 billion total page views per month
- 58.9 million unique devices per day
- More than 13k new editors each month
- More than 75k active editors month-to-month

About: Wikipedia

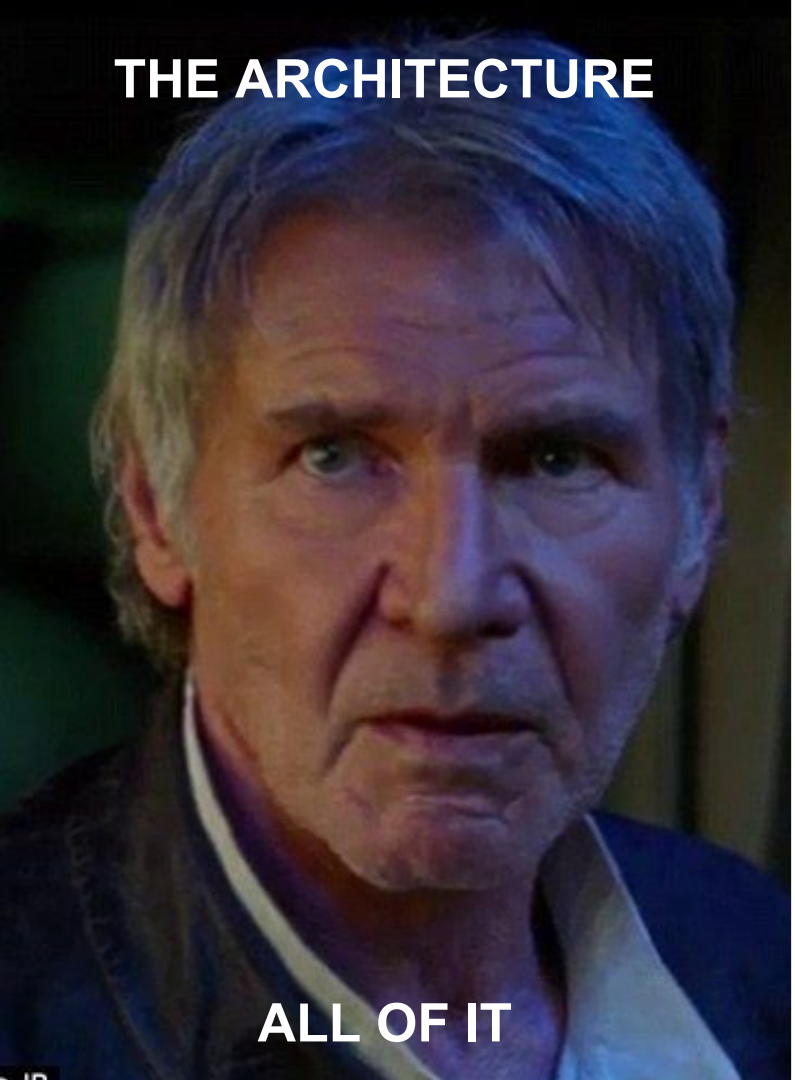
- More than 38 million articles in 290 languages
- Over 10k new articles added per day
- 13 million edits per month
- Ranked #6 globally in web traffic

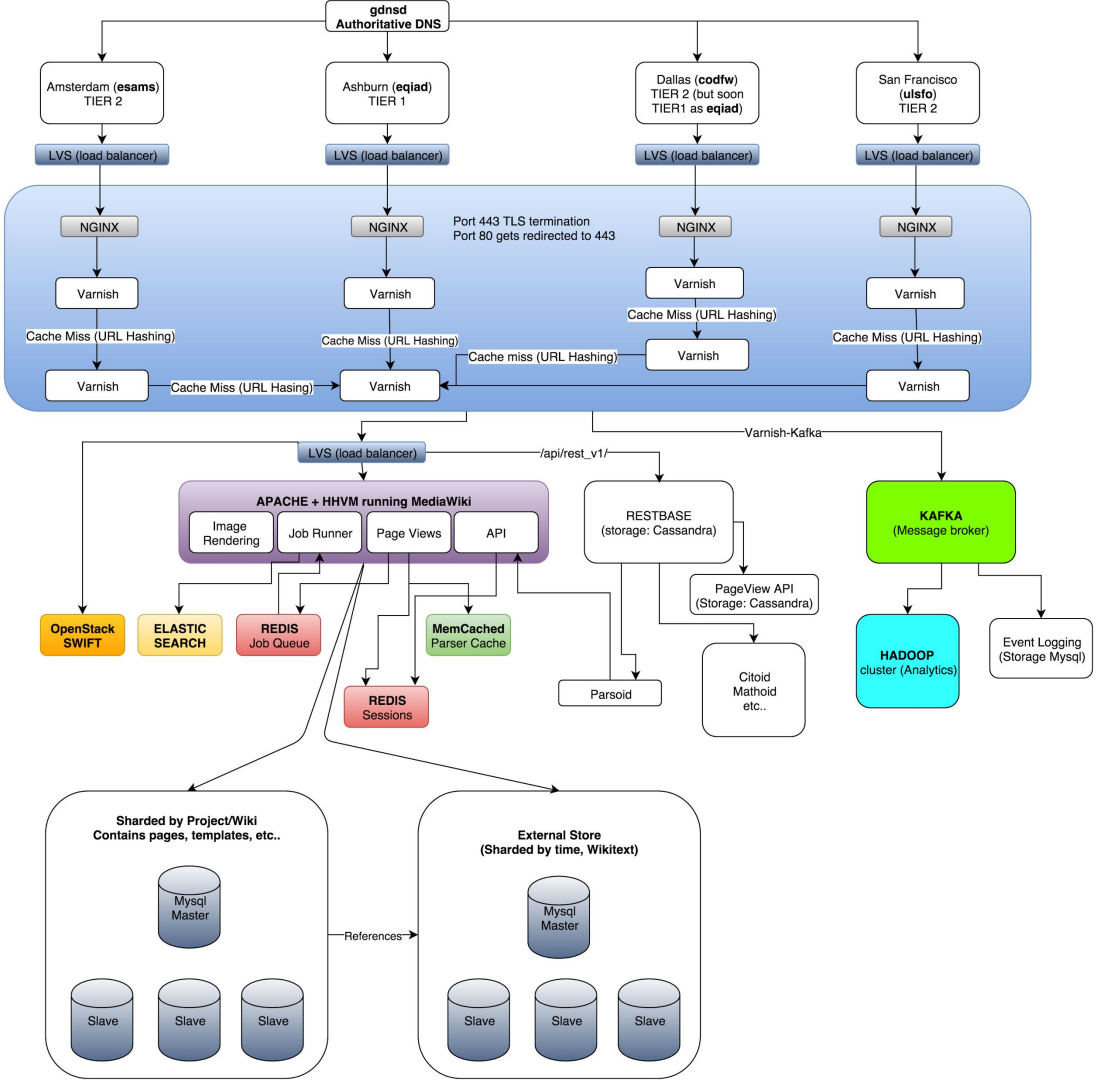
Wikimedia Architecture

LAMP



THE ARCHITECTURE






visual
editor

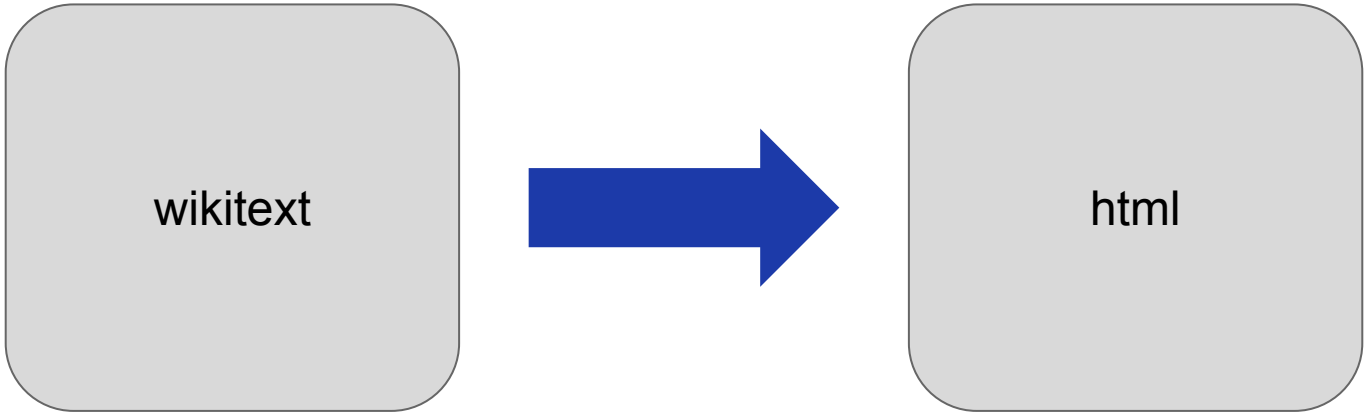
HTML

```
<h1>
  Star Wars: The Force Awakens
</h1>

<p>
  Star Wars: The Force Awakens is a 2015 American epic space opera
  film directed, co-produced, and co-written by
  <a href="/wiki/J._J._Abrams" title="J. J. Abrams">
    J. J. Abrams
  </a>
</p>
```

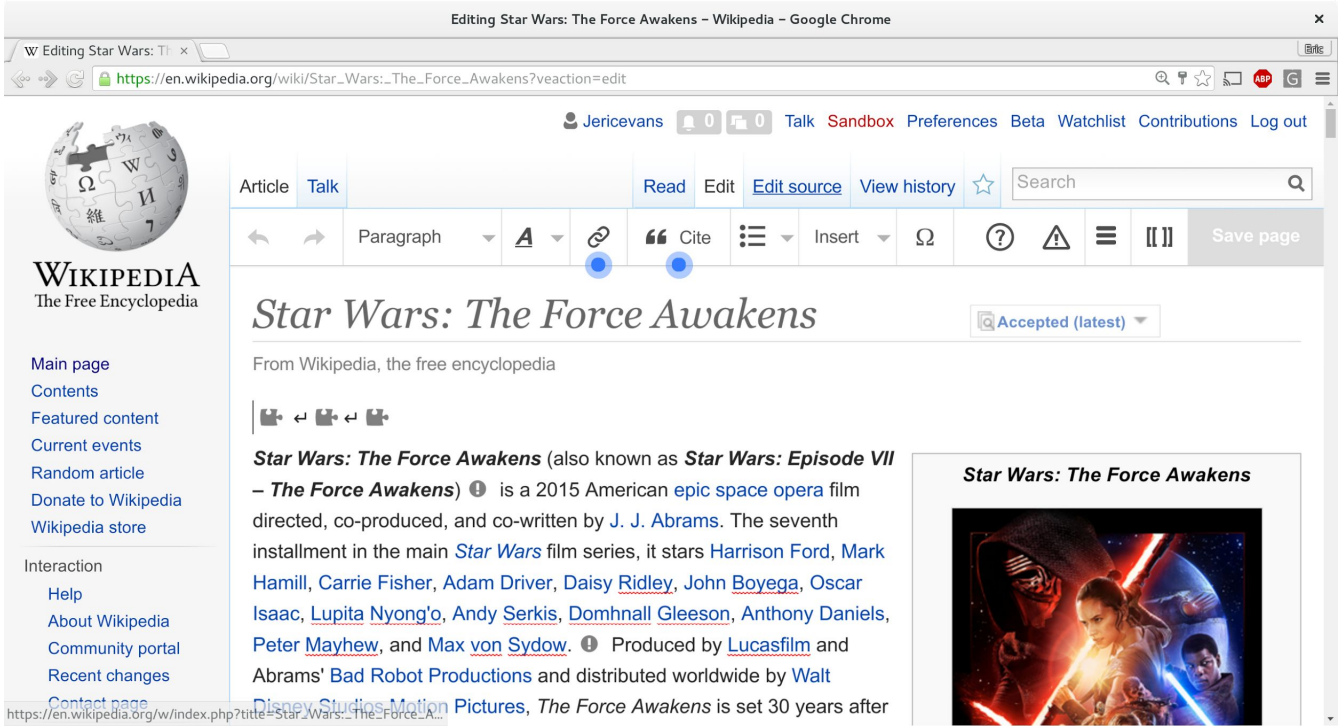
Wikitext

```
= Star Wars: The Force Awakens =

Star Wars: The Force Awakens is a 2015 American epic space opera
film directed, co-produced, and co-written by [[J. J. Abrams]].
```

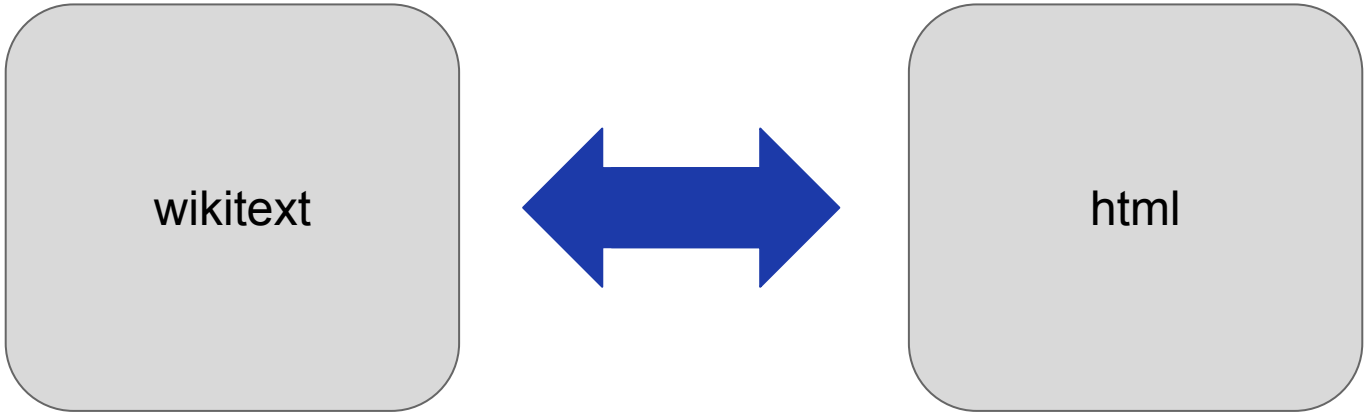
Conversion



WYSIWYG


Editing Star Wars: The Force Awakens – Wikipedia – Google Chrome
https://en.wikipedia.org/wiki/Star_Wars:_The_Force_Awakens?veaction=edit
Jericevans 0 0 Talk Sandbox Preferences Beta Watchlist Contributions Log out
Article Talk Read Edit Edit source View history Search
Paragraph Cite Insert Save page
WIKIPEDIA
The Free Encyclopedia
Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store
Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page
Star Wars: The Force Awakens
Accepted (latest)
From Wikipedia, the free encyclopedia
Star Wars: The Force Awakens (also known as Star Wars: Episode VII – The Force Awakens) is a 2015 American epic space opera film directed, co-produced, and co-written by J. J. Abrams. The seventh installment in the main Star Wars film series, it stars Harrison Ford, Mark Hamill, Carrie Fisher, Adam Driver, Daisy Ridley, John Boyega, Oscar Isaac, Lupita Nyong'o, Andy Serkis, Domhnall Gleeson, Anthony Daniels, Peter Mayhew, and Max von Sydow. Produced by Lucasfilm and Abrams' Bad Robot Productions and distributed worldwide by Walt Disney Studios Motion Pictures, The Force Awakens is set 30 years after
Star Wars: The Force Awakens
https://en.wikipedia.org/w/index.php?title=Star_Wars:_The_Force_A...

Conversion



Character-based diffs

Metadata

```
[[Foo|{{echo|bar}}]]
```

```
<a rel="mw:WikiLink" href="./Foo">
  <span about="#mwt1" typeof="mw:Object/Template"
        data-parsoid="{...}" >bar</span>
</a>
```

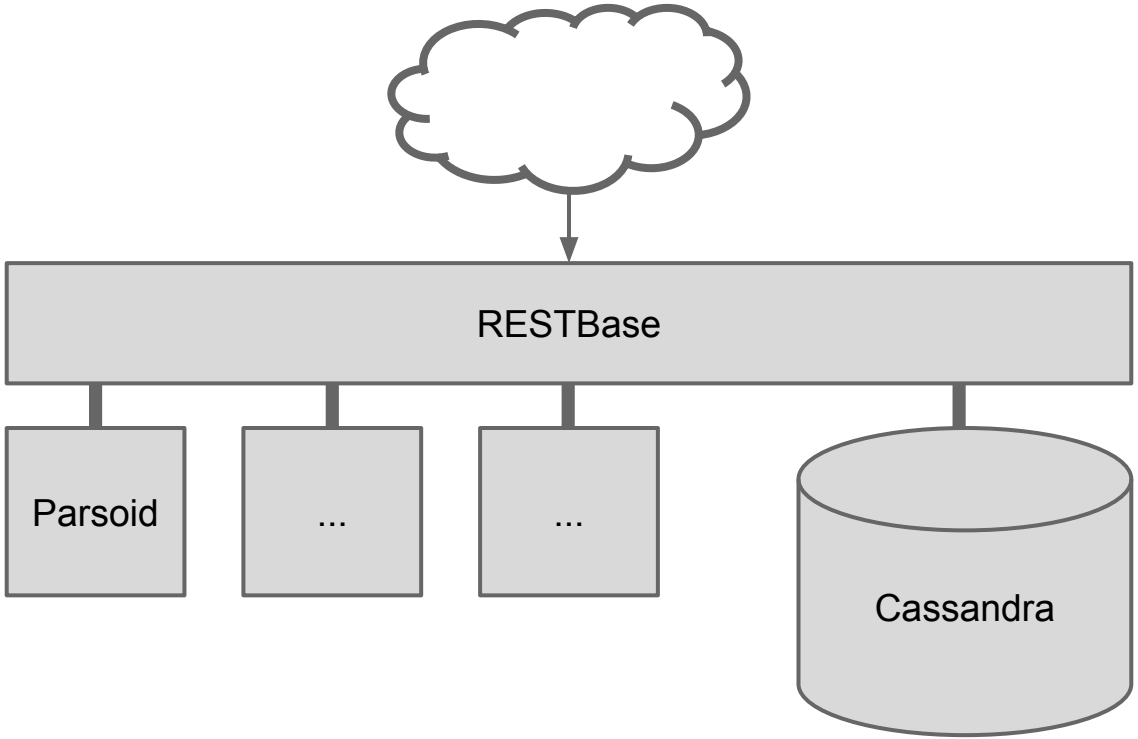
Parsoid

- Node.js service
- Converts wikitext to HTML/RDFa
- Converts HTML/RDFa to wikitext
- Semantics, *and* syntax (avoid dirty diffs)!
- Expensive (slow)
- Resulting output is large

RESTBase

- Services aggregator / proxy (REST)
- Durable cache (Cassandra)
- Wikimedia’s content API (e.g. https://en.wikipedia.org/api/rest_v1?doc)

RESTBase



Other use-cases

- Mobile content service
- Math formula rendering service
- Dumps
- ...

Cassandra

Environment

- Cassandra 2.2.6
- 2 datacenters
- 3 ~~racks~~ rows per datacenter
- 6 replicas (3x3)
- 18 hosts (16 core, 128G, SSDs)
- 54 Cassandra nodes
- Deflate compression (~14-18%)
- Read-heavy workload (5:1)

Data model

key-revision-value

- Key
  - Site domain + document title
  - Used as partition key (determines cluster distribution)
- Revision
  - Monotonically increasing integer assigned by MediaWiki
  - Many-to-one relationship with key
- Render
  - TimeUUID assigned by RESTBase
  - Many-to-one relationship with revision
- Value
  - Blob

As CQL DDL

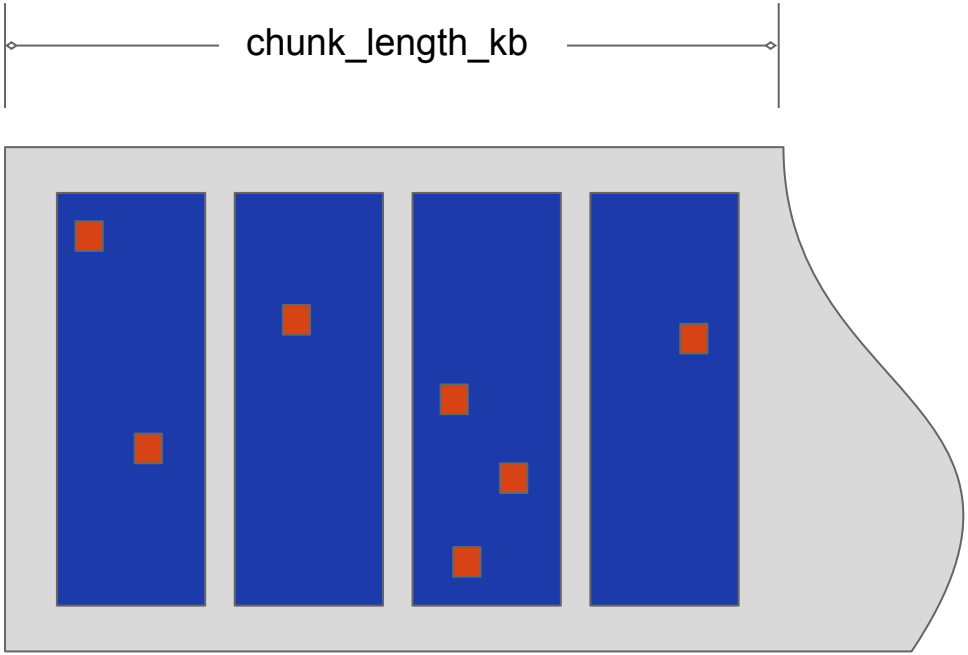
```
CREATE TABLE data (
    domain  text,
    title   text,
    rev     int,
    render  timeuuid,
    value   blob,
    PRIMARY KEY ((domain, title), rev, tid)
) WITH CLUSTERING ORDER BY (rev DESC, tid DESC)
```

Data model

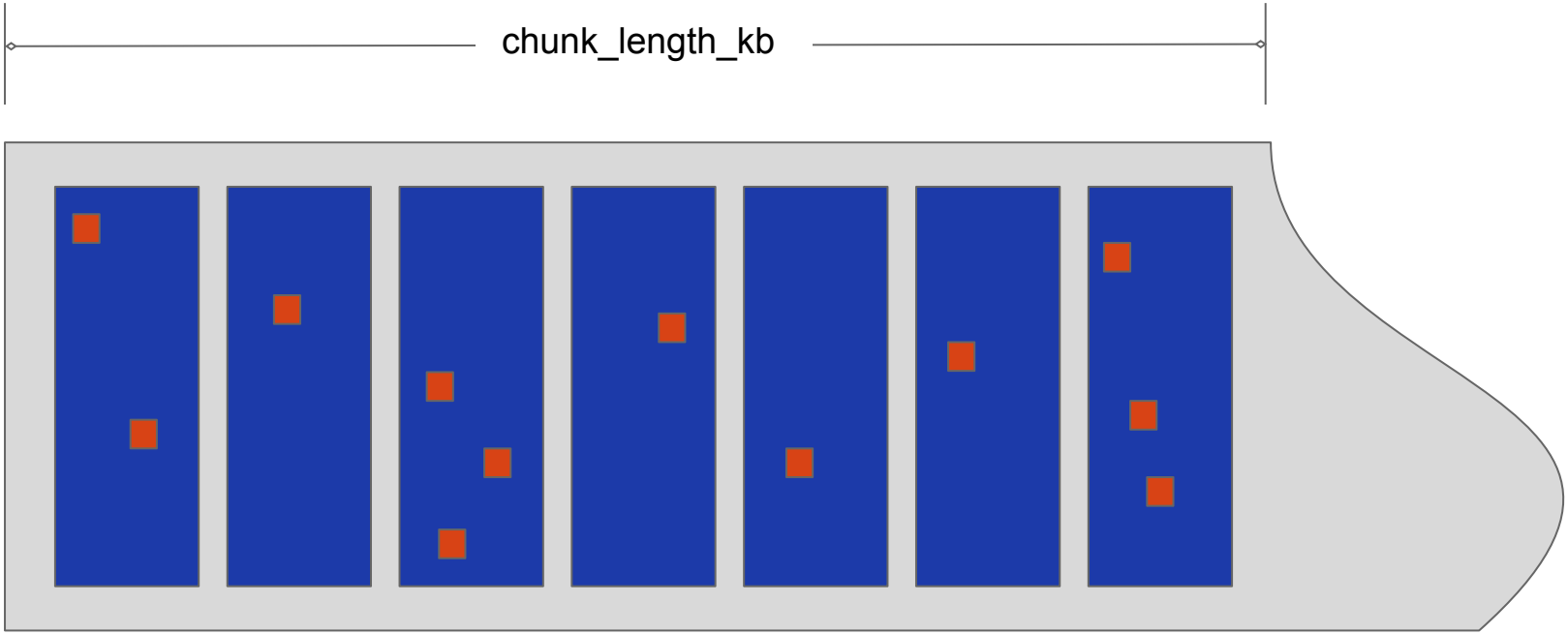

(en.wikipedia.org, Star_Wars:_The_Force_Awakens)
717862573
717873822
...
1f2dd66c...7c7a913d3d8a
97466b12...7c7a913d3d8a
...
827e2ec2...7c7a913d3d8a
bdebc9a6...7c7a913d3d8a
09877568...7c7a913d3d8a

Compression

Compression



Compression



Brotli compression

- Brought to you by the folks at Google; Successor to deflate
- Cassandra implementation (https://github.com/eevans/cassandra-brotli)
- Initial results very promising
- Better compression, lower cost (apples-apples)
- And, wider windows are possible (apples-oranges)
  - GC/memory permitting
  - Example: `level=1`, `lgblock=4096`, `chunk_length_kb=4096`, yields 1.73% compressed size!
  - https://phabricator.wikimedia.org/T122028
- Stay tuned!

Compaction

Compaction

- The cost of having log-structured storage
- Asynchronously optimize data on disk for later reads
- At a minimum, reorganize into fewer files
- If possible, reorganize to position query results nearer each other
- Garbage collect
  - Remove overwritten values
  - Expiring TTLs
  - Removing deleted (aka tombstoned) data (after a fashion

Compaction strategies

- Size-tiered
  - Combines tables of similar size
  - Oblivious to column distribution; Works best for workloads with no overwrites/deletes
  - Minimal IO
- Leveled
  - Small, fixed size files in levels of exponentially increasing size
  - Files have non-overlapping ranges within a level
  - Very efficient reads, but also quite IO intensive

Compaction strategies

- Size-tiered
  - Combines tables of similar size
  - Oblivious to column distribution; Works best for workloads with no overwrites/deletes
  - Minimal IO
- Leveled
  - Small, fixed size files in levels of exponentially increasing size
  - Files have non-overlapping ranges within a level
  - Very efficient reads, but also quite IO intensive

Nope.

- Too much disk IO
- Too much GC throughput
- Way too much disk IO

Compaction strategies

- Size-tiered
  - Combines tables of similar size
  - Oblivious to column distribution; Works best for workloads with no overwrites/deletes
  - Minimal IO
- Leveled
  - Small, fixed size files in levels of exponentially increasing size
  - Files have non-overlapping ranges within a level
  - Very efficient reads, but also quite IO intensive
- Date-tiered
  - For append only, total ordered data
  - Avoids mixing old data with new
  - Cold data eventually ceases to be compacted

Nope.

- Too difficult to reason about
- Optimizations easy defeated

Compaction strategies

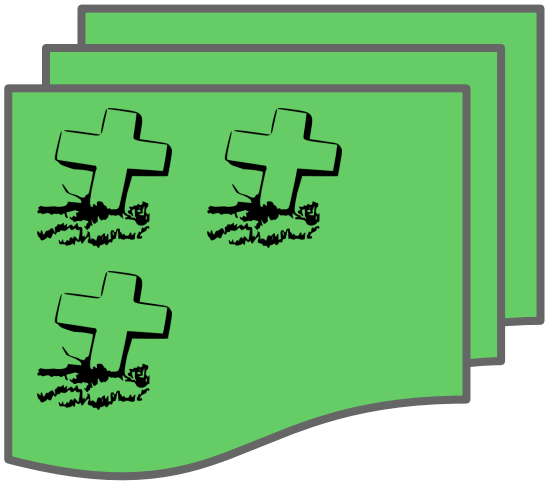
- Size-tiered
  - Combines tables of similar size
  - Oblivious to column distribution; Works best for workloads with no overwrites/deletes
  - Minimal IO
- Leveled
  - Small, fixed size files in levels of exponentially increasing size
  - Files have non-overlapping ranges within a level
  - Very efficient reads, but also quite IO intensive
- Date-tiered
  - For append only, total ordered data
  - Avoids mixing old data with new
  - Cold data eventually ceases to be compacted

But remember: Compaction is also for...

- Removing overwritten values
- Expiring TTLs
- Removing deleted (aka tombstoned) data (after a fashion)

The Overlapping Tables Problem

Tables being compacted



Tables not being compacted

Garbage Collection (JVM)

G1GC

- Early adopters of G1 (aka “Garbage 1st”)
- Successor to Concurrent Mark-sweep (CMS)
- Incremental parallel compacting collector
- More predictable performance than CMS

Humongous objects

- Anything >= ½ region size is classified as *Humongous*
- Humongous objects are allocated into *Humongous Regions*
- Only one object for a region (wastes space, creates fragmentation)
- Until 1.8u40, humongous regions collected only during full collections (Bad)
- Since 1.8u40, end of the marking cycle, during the cleanup phase (Better)
- Treated as exceptions, so should be exceptional
  - For us, that means 8MB regions
- Enable GC logging and have a look!

Node density

Motivation

- Compaction
- GC
- Failure scenarios
- ...

What we do

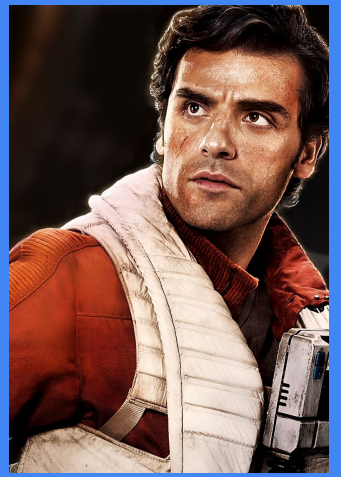
- Processes (yup)
- Puppetized configuration
  - `/etc/cassandra-a/`
  - `/etc/cassandra-b/`
  - systemd units
  - Etc
- Shared RAID-0

What we should have done

- Virtualization
- Containers
- Blades
- Not processes

The Good

- Fault tolerance
- Availability
- Datacenter / rack awareness
- Visibility
- Ubiquity
- Community



The Bad

- Usability
  - Compaction
  - Streaming
  - JMX
- Vertical Scaling

The Ugly

- Upgrades
- Release process