Create account Log in

Page Discussion Read View source View history Search Wikitech

Incident documentation/20160505-ChangeProp RESTBase Parsoid

< Incident documentation

Contents [hide]

Summary

A combination of several subtle issues lead to a feedback loop in a test deploy of the ChangeProp service, which is designed to orchestrate the propagation of changes across services using EventBus. This created a spike of un-cacheable requests to RESTBase and Parsoid, which temporarily overloaded those clusters, and resulted in users experiencing high latencies and request errors for a duration of about one hour.

Timeline

- 2016-05-04T15:02:45 first requests to parsoid for oldId 106801025 (from ChangePropagation/WMF)
- <date change>
- 2016-05-05T02:16:10 first INFO [HintedHandoff:2] 2016-05-03 02:16:10,523 HintedHandOffManager.java:486 - Timed out replaying hints to /10.192.32.137; aborting (25847 delivered) (on restbase1014)
- 2016-05-05T02:28 org.apache.cassandra.db.compaction.CompactionTask log events elevated
- 2016-05-05T02:3X bytes_out and cpu_user start to spike on rb cluster
- 2016-05-05T02:3X load spikes on parsoid cluster
- 2016-05-05T02:34:16 first "Retry count exceeded" error in /srv/log/changeprop/main.log (for http://fr.wikipedia.org/api/rest_v1/page/html/%EA%9D%AE)
- 2016-05-05T02:38:14 first alert: <icinga-wm> PROBLEM - restbase endpoints health on restbase1013 is CRITICAL: CHECK_NRPE: Socket timeout after 10 seconds.
- 2016-05-05T03:20:00 (ca.) Ori stops changeprop service
- 2016-05-05T03:30:00 (ca.) Load on Parsoid and RESTBase drops, service recovers.

Analysis

1. The lower-case version of https://fr.wikipedia.org/api/rest_v1/page/html/%EA%9D%AE was actually an illegal title according to the wiki's title capitalization setting, but this was not detected as a side-effect of an mbstring / HHVM bug. When interpreted according to unicode & normalization rules of the wiki, it was a redirect to itself.
2. The uppercase title was edited, and change-propagation picked up the edit and requests updates in restbase with a no-cache request for the page summary end point.
   - Note: The same update is also triggered by normal job processing.
3. RESTBase checked whether the page is a redirect. While the uppercase version of the page was not a redirect, the lower-case version was. RESTBase always stores content and revision information at normalized titles, which for this wiki means an upper-cased first char. The lower-case version of the article happened to have a higher revision number, so the redirect flag was picked up. To find the redirect target,

Main page
Recent changes
Server admin log (Prod)
Server admin log (RelEng)
Deployments
SRE/Operations Help
Incident status

Cloud VPS & Toolforge
Cloud VPS documentation
Toolforge documentation
Request Cloud VPS project
Server admin log (Cloud VPS)

Tools
What links here
Related changes
Special pages
Permanent link
Page information
Cite this page

Print/export
Create a book
Download as PDF
Printable version

the access / redirect filter then requested HTML to parse redirect target, but forwarded all the headers, including no-cache.

4. In parsoid.js (in the RESTBase codebase), the missing revision parameter then triggered no-cache revision fetching, which retrieved the latest revision of the uppercase title from MW API, yielding 106801025.
5. This revision was then requested from Parsoid. The content returned from Parsoid was classified as differing & was saved.
   - Reason for always-diff: The contentReq at https://github.com/wikimedia/restbase/blob/master/sys/parsoid.js#L420 was only based on the request parameters, which in this case did not include the revision. The latest stored revision was actually 125284517 (corresponding to the lowercase title), thus causing the content to mismatch when compared to the re-render of 106801025.
   - Secondary updates emitted a resource change event for HTML
6. Changeprop picked up the resource change event from #5, going back to #1 of the loop.

This by itself would have just caused a sequential repeat of a single request, and would not have lead to an outage. However, several other issues caused an exponential retry amplification, which in turn created enough load on RESTBase and Parsoid to make this an outage:

- RESTBase secondary updates (temporarily inline) also perform an update request to the Mobile Content Service (MCS)
  - MCS in turn requests HTML from RESTBase without no-cache but with ?redirect=false, gets HTML with self-redirect
  - Inline redirect extraction was still active in MCS, causing it to repeat requests 10x (all served from storage) until redirect limit reached, then returned 400 from parsoid-access.js (lots of log entries for this in RB logs).
- Since this was a no-cache request, secondary updates are done synchronously, and errors are reported. A generic 500 was returned from RESTBase.
- The change-propagation service received the 500 response (triggered by MCS redirect limit) and, after a delay of ~500ms, retried 5x until the retry count was exceeded.
- ChangeProp processed retries eagerly & in parallel with resource_change events. There was no limit on processing concurrency apart form CPU, and delays for retries were short (500ms). As a result, parallelism of retries & event processing ramped up until the CPU was saturated, sending out at least 500+ un-cacheable requests per second.

Actionables

The progress and discussion on actionables is tracked in phab:T134537.

RESTBase

- Don't propagate the cache-control header for access checks: https://github.com/wikimedia/restbase/pull/604
- change contentReq at https://github.com/wikimedia/restbase/blob/master/sys/parsoid.js#L420 to always use a / the same revision as is used for the parsoid request
- look into stopping no-cache revision fetching as restriction update mechanism (avoids mismatch between revision & content storage)
- Improve logging / error reporting from temporary dependency updates

ChangeProp service

- Implement concurrency limiting on change propagation & enforce conservative limits and delays
- Change Propagation should not follow redirects

Mobile Content Service

- Remove redirect follow code from MCS

Parsoid

- investigate slow render of the pages involved here
- consider setting up rate limits for repeat parses of the same domain / title / revision: Ban a set of titles

Alerts

- Create functional cluster checks for all services (and have them page!)

Category: Incident documentation

This page was last edited on 9 May 2016, at 11:46.