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

Incident documentation/20200126-app server latency

< Incident documentation

document status: draft

Contents [hide]

Summary

Note: the corresponding maps outage is not treated here. Its timings will just be reported in the timeline for reference.

At 19:47 UTC, the news of Kobe Bryant's death was announced. This caused a surge in both edits and page views, causing a stampede of requests and a general slowdown of both the application servers and the apis. This caused both high contention in editing/parsing as expected, and can be seen by the number of poolcounter full queues, as well as high cpu usage, very high response times, and general unavailability of the MediaWiki application layer (both appservers and api). This was caused by a series of concauses:

- High contention in editing/parsing, resulting in a lot of workers blocked on waiting for poolcounter.
- Saturation of the network link of two memcached servers due to the high edit/reparse activity. This caused about 1/9th of the memcached keys to become unavailable
- A spike in latencies resulting from the failure of the memcached servers caused a rise in the cache-misses for babel calls from the appservers, flooding the api servers. pretty much as the same mechanism of the incident of February 4 (although the ignition of the incident was different).

Given the sustained higher level of requests, it took more than one hour for the clusters to recover.

Number of full queues on poolcounter during the incident.

Impact

1 hour of almost complete unavailability of the backend servers. During that period, a large number of requests to our sites will have timed out or got an error response.

As can be seen from the graph of all requests, the average request rate was just barely under what the normal rate is, but we can estimate there was a large number of requests that we weren't able to answer to. Almost 90% of the backend requests from ATS failed.

Successful responses from all of our caching frontends during the incident.

Detection

The monitoring correctly and promptly detected the issue, and 3 minutes later we received the first page. Given most SRE were already around for a previous incident, most of us didn't need the page to be online.

```
2020-01-26 19:52:33     +icinga-wm      PROBLEM - MediaWiki memcached error rate
on icinga1001
2020-01-26 19:53:57     +icinga-wm      PROBLEM - High average GET latency for
mw requests
2020-01-26 19:55:11     +icinga-wm      PROBLEM - LVS HTTPS IPv4 #page on text-
lb.codfw.wikimedia.org
```

Timeline

All times in UTC.

- 19:50 Mcrouters on almost every host report failures connecting to mc1019 (mostly) and other servers. **OUTAGE BEGINS**
- 19.51 Latency spikes up on the appserver cluster. Cache hit ratio for Babel requests is reduced to about 25% of normal
- 19.52 Investigation begins
- 19.53 Latency spikes up on the api cluster too.
- 20.01 Cause is tentatively identified as a side effect of the news spreading and an editing stampede happening
- 20.08 The largest part of the memcached errors are recovered, but the latencies are still very high (p75 for the appservers at precisely 10 seconds)
- 20:13 Babel calls are identified as a possible cause.
- 20:16 Developers are called in order to try to understand what is going on. Given it's sunday and most people are flying to All-Hands, this takes some time.
- 20:30 Attention also moves to other pages doing cross-cluster requests like `ImagePage::printSharedImageText`, and to the fact we're reaching the api for babel via the caching layer
- 20:35 Various mitigations are proposed - including setting `fetchDescription => false` in wmf-config/filebackend.php
- 20:35 Kartotherian reaches 100% cpu utilization, starts intermittently paging
- 20:41 It is noticed that some varnish-fes are constantly crashing
- 20:43 An anomalous amount of 302s are noticed being returned from the application servers. Upon investigation, those are shown to be mostly the Apple search interface searching for Kobe Bryant.
- 21:01 It gets decided to rollout a very aggressive timeout in MediaWiki core's HttpRequest request abstraction.
- 21:04 Most services recover before the fix is deployed. **OUTAGE ENDS**
- up to 24:00 - cpu on the maps cluster is still constantly at 100%.

Conclusions

What went well?

- Monitoring detected the outage quickly

What went poorly?

- We weren't able to resolve the issues or identify the origin of the problem ourselves. The issue self-resolved.
- While we identified a few possible causes of the issue, we didn't prepare any fix before the outage was over.

Where did we get lucky?

- We were aware of the news that caused the outage; this made the whole event less surprising
- The outage resolved itself likely when the overall traffic waned.

How many people were involved in the remediation?

- 8 SREs, 4 developers (2 WMF, 1 WMDE, 1 volunteer)

Actionables

- Reduce Wikibase calls to Babel when not needed https://phabricator.wikimedia.org/T243725 ✓ **Done**
- Reconsider https://gerrit.wikimedia.org/r/#/c/operations/puppet/+/511751/ (apache forensic logging)
- Proxy calls from MediaWiki to all https calls through a proxy https://phabricator.wikimedia.org/T244843

- Reduce read pressure on memcached servers by adding a machine-local Memcache instance https://phabricator.wikimedia.org/T244340
- Investigate why a slowdown in response times from the API causes a surge in cache misses for Babel data in Wikibase. See https://phabricator.wikimedia.org/T244877

Categories: Incident documentation | Incident documentation drafts