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/20200319-parsercache

< Incident documentation

document status: in-review

Contents [hide]

Summary

Parsercache databases got overloaded due to a malfunctioning host which resulted on spikes of connections on the other 2 active hosts and increased latency on our mwapps servers.

Impact

- Query latency was increased https://grafana.wikimedia.org/d/RIA1lzDZk/application-servers-red-dashboard?orgId=1&var-datasource=eqiad%20prometheus%2Fops&var-cluster=appserver&var-method=GET&var-code=200&from=1584378662200&to=1584387599259&fullscreen&panelId=31
- mw app servers got their workers saturated: https://grafana.wikimedia.org/d/000000550/mediawiki-application-servers?

orgId=1&fullscreen&panelId=92&from=1584358493597&to=1584421429638
- Higher than usual response time https://grafana.wikimedia.org/d/RIA1lzDZk/application-servers-red-dashboard?orgId=1&var-datasource=eqiad%20prometheus%2Fops&var-cluster=appserver&var-method=GET&var-code=200&from=1584378662200&to=1584387599259&fullscreen&panelId=9

Detection

Icinga paged for pc1008 host that was having performance degradation

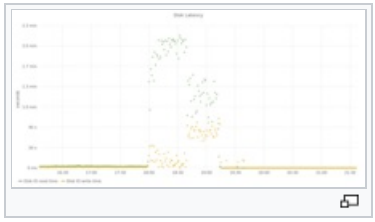
```
18:43:14 <+icinga-wm> PROBLEM - MariaDB Slave SQL:
pc2 #page on pc1008 is CRITICAL: CRITICAL
slave_sql_state could not connect
https://wikitech.wikimedia.org/wiki/MariaDB/trouble
shooting%23Depooling_a_slave
```

Timeline

All times in UTC.

- 18:00 **Degradation begins**

- 18:00 pc1008 starts having performance issues and its disk latency starts increasing, connections start to pile up on pc1008
- 18:00 Other hosts (pc1007 and pc1009) also start suffering more idle connections as the result of pc1008 failing to handle connections as fast as usual
- 18:00 Average response time increases
- 18:43 *<+icinga-wm> PROBLEM - MariaDB Slave SQL: pc2 #page on pc1008 is CRITICAL: CRITICAL slave_sql_state could not connect*
- 18:43-19:44 A number of SREs and 2 DBAs respond and troubleshooting starts
- 19:11 DBAs Replace pc1008 with pc1010 (which is a spare for a different pc group, and has 1/3 of the key), but worth trying as there were no more ideas and pc1008 was checked for HW errors, misconfigurations and such and all looked fine anyways.
- 19:12 Response time, idle connections on other hosts, latency...they all start to get better
- 19:24 Values almost around the same before the incident (considering that 1/3 of the pc keys were gone)
- 19:24 **Degradation stops**



Conclusions

The hardware performance degradation was hard to detect via the usual checks: broken BBU, degraded RAID, disks with errors that hasn't removed from the RAID, memory issues.... As nothing appeared to be broken, DBAs didn't consider pc1008 as the core of the issue. The fact that all the parsercache showed similar connections spike pattern made us think that the problem was on the other side of the spectrum (MW).

We later learned thanks to Brad, that parsercache has a "double write" behaviour we didn't know of and if one of those fails, the others keep hanging until the request is processed or shutdown.

What went well?

- When we planned the parsercache refresh a year ago, we decided to buy a host to have it as a spare, precisely for these kind of situations.

What went poorly?

- DBAs were not aware of this parsercache behaviour so they didn't consider pc1008 affecting other host as a possibility (later explained by Brad on https://phabricator.wikimedia.org/T247788#5975667%7CT247788#5975667):

```
Each write to ParserCache sets two keys into the backend, which will probably
get sharded to two different servers. Once SqlBagOStuff opens a connection to
one of the servers, it keeps
that connection open until request shutdown. So if we assume that pc1008 is
somehow failing in a way that has connections hang open for a while, we'd also
see a smaller increase in
idle open connections on pc1007 and pc1009 for the cases where ParserCache's
first write goes to pc1007/pc1009 and the second one goes to pc1008. That seems
consistent with what the
three graphs show.
```

- Trying to get ahold of CPT via IRC wasn't possible.
- The hardware degradation pc1008 had, was hard to detect and was only detected a day after, with lots of testing (https://phabricator.wikimedia.org/T247787#5975506)

Where did we get lucky?

- Just to try things, we decided to replace pc1008 with pc1010 but without much expectations and it worked

How many people were involved in the remediation?

- 2 DBAs
- 3 SREs
- 2 WMDE Devs

Links to relevant documentation

This explanation by Brad resumes what was happening from MW side https://phabricator.wikimedia.org/T247788#5975667 and https://phabricator.wikimedia.org/T247788#5976651

Actionables

- [RFC] improve parsercache replication, sharding and HA: https://phabricator.wikimedia.org/T133523
- Investigate pc1008 for possible hardware issues / performance under high load: https://phabricator.wikimedia.org/T247787
  - Once pc1008 is back full - repool it to make sure it is fully fixed after re-creating the raid
  - Purge pc1010 old rows once it is out of rotation
- Parsercache sudden increase of connections: https://phabricator.wikimedia.org/T247788#5976651

Categories: Incident documentation in-reviews | Incident documentation