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/20140619-parsercache

< Incident documentation

Contents [hide]

Summary

There was a site outage and 5xx spike between 2014-06-19 19:07:32 and ~19:20 caused by pc1002 going down. Basically, one of three parsercache boxes was overwhelmed and everything else turned out to be a house of cards.

Timeline

- Trimmed and annotated IRC log.
- PDF of incinga graphs for pc1002 before/during the outage.
- pc1002 bytes in
- Parser Cache mass eviction
- Spot the puppet runs

Rough outline:

- A mass of "Too many connections" errors for pc1002 in dberror.log
- A bunch of mw nodes went critical in Icinga
- pc1002 went critical in Icinga
- mutante observed an OOM error on the pc1002 console
  - RobH observed no ssh/ping and so powercycled pc1002
- mutante found mysqld failed to restart after powercycle due to a pid file
- After restart and MariaDB recovery the site came back up.

Conclusions

What weakness did we learn about and how can we address them?

1. All three pc100[123] boxes encountered a spike in traffic. Note that this was write-traffic; pc1001 and pc1003 both stayed alive and tendril logged a spike in db writes. Robla noticed enwiki Template:Navbox was edited immediately prior. This may account for the spike, but it's important to note that parser cache sees such spikes from time to time and /doesn't die/, so traffic was a contributing factor but *not* the whole problem.

2. pc1002 ganglia logged a significant spike in bytes_in[1] and wait_io in addition to the load from the traffic spike, compared to pc1001 and pc1003. The syslog shows that puppet ran at that time and took longer than normal, likely due to the already heavy mysqld load. The extra bytes_in was presumably puppet traffic and the extra wait_io potentially already-heavy DB activity plus puppet (plus swap? unknown considering oom/panic). So puppet running was the final trigger but also *not* the whole problem.

3. pc1002 evidently didn't die cleanly and was still responding to at least some mysqld tcp connections for at least 1min before the outage registered and ssh/ping demonstrably died. No exact timestamp available for the OOM and/or kernel panic mutante saw. For some reason Mediawiki LB did not simply stop using pc1002 but instead locked up waiting for it until it rebooted. Tim theorised transactions waiting to commit, which matches up with transactions rolled back during InnoDB recovery. I don't know if anyone got a chance to check the state of tcp connections from an MW node during the incident (if anyone responds to something like this in the future please

also grab lsof -i tcp output from app servers)?

Proposals

1. Add more parsercache hardware. This is potentially necessary but I don't think it should be the only message we take away from this incident.

2. A puppet run should not start if a box is under abnormal load. We brought this possibility up last week in #mediawiki_security because I observed a couple of recent DB load spikes on regular S[1-7] slaves that corresponded with puppet causing high wait_io. db1021 specifically had an almost identical max_connections incident but thankfully not during a significant traffic spike. Is it possible we don't realize the impact of puppet destabilizing other systems too, besides the DBs?

3. Mediawiki PHP may need some better way of handling a DB host that is flaky rather than completely down. Historically we've seen similar lock-up behavior on S[1-7] where one slave having problems leads to unnecessary outages. As it happens this week we discussed options for DB proxies (haproxy probably) in #mediawiki_security, both for HA and maintenance reasons. It's possible that PHP simply should not be connecting directly to databases without hand-holding. This need to take Mediawiki LB and query groups into account. May even need heartbeat and STONITH?

4. Long term, parsercache really doesn't need to be a full RDBMS. I gather originally it was done because InnoDB with a big buffer is a good trade-off between performance and durability. But the queries are all simple and presumably could map to a simpler datastore able to handle higher throughput with less overhead and better connection characteristics.

Actionables

- Status: **Declined** - Run puppet nice'd
  - RT 9
- Status: **Declined** - A puppet run should not start if a box is under abnormal load.
  - RT 7888
- Status: **Declined** - Improve how Mediawiki handles a DB host that is flaky rather than completely down
  - bugzilla:68062
- Status: **Declined** - Migrate parsercache away from being a full RDBMS.
  - RT 7889

Category: Incident documentation