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/20190723-network eqiad

< Incident documentation

document status: in-review

Contents [hide]

Summary

A fiber connecting racks A6 and A7 in asw2-a-eqiad's virtual switch chassis somehow became damaged during work in the datacenter, and [speculation] caused a high bit error rate, leading to lots of packet loss, TCP retransmits, and application-level errors - most prominently to mediawiki talking to memcached via mcrouter. This had a secondary fallout in making logstash unavailable, as Mediawiki generated a ton of logs reporting memcached/mcrouter errors.

Impact

The difficulty in communicating with memcached (rather than the cache being completely unavailable) caused multiple appservers to slow down to the point that requests piled up on them. Approx 25k HTTP 50X served as a result. https://logstash.wikimedia.org/goto/2e16490fd1695f799e15b63ad38be150

Additionally, Mediawiki ResourceLoader latency increased by ~10-15ms at 75%ile, ~750ms at 95%ile, ~30s at 99%ile, as shown by the performance metrics..

Detection

Detection was indirect, and thus quite inefficient.

- Asw2-a-eqiad was detected going down briefly, but no further issue was reported
- Multiple application servers started reporting rendering issues well within the issue window
- Availability alerts were raised during the peak of the issue
- logstash became unavailable due to the huge intake of MediaWiki error messages, and paged

the issues were properly identified once they became user-visible though.

Timeline

All times in UTC.

- ~18:00 - Start of prep work (e.g. depools, poweroffs) for Rack A7 power maint: https://phabricator.wikimedia.org/T227143
- ~18:03 - Memcache CAS requests spike up dramatically

Around the same time, the first timeouts appear. The two values are interlocked because timeouts and memcached failures in general cause a CAS read to be performed by the MediaWiki driver

- 18:17 - `<+icinga-wm> PROBLEM - Host ps1-a7-eqiad is DOWN: PING CRITICAL - Packet`

`loss = 100%` Probably start of maintenance.

- 18:29 - 18:30 `aws2-a-eqiad` flaps
- 18:30 - 18:43 first small burst of mcrouter TKOs and timeouts
- 18:33 - Rob notes the asw2-a-eqiad flap was "not expected", proceeds to kill power on one of the two sides in a7
- The switch logs confirm the action

```
   Jul 23 18:34:43  asw2-a-eqiad alarmd[2039]: Alarm set: PS SFXPC color=YELLOW,
class=CHASSIS, reason=FPC 7 PEM 1 is not powered
   Jul 23 18:35:59  asw2-a-eqiad alarmd[2039]: Alarm cleared: PS SFXPC
color=YELLOW, class=CHASSIS, reason=FPC 7 PEM 1 is not powered
```

- 18:42 - Rob notes side A is done, and that work is moving to side b. Again, switch logs seem to confirm:

```
   Jul 23 18:43:34  asw2-a-eqiad alarmd[2039]: Alarm set: PS SFXPC color=YELLOW,
class=CHASSIS, reason=FPC 7 PEM 0 is not powered
   Jul 23 18:44:14  asw2-a-eqiad alarmd[2039]: Alarm cleared: PS SFXPC
color=YELLOW, class=CHASSIS, reason=FPC 7 PEM 0 is not powered
```

- 18:51 - mcrouter TKOs and timeouts begin in earnest (peaking at ~600k TKOs/sec) **OUTAGE BEGINS**
- 19:10 - First page (logstash failure, secondary fallout from error ingestion)
- 19:27 - Memcached errors from mediawiki are localized in rack A7 (memcached) / mostly in rack A6 (appservers). Thus appservers in A6 get depooled, to no avail.
- 19:29 - TCP retransmits (esp. for Analytics) skyrocket, as does network traffic
- 19.45 - The problem seems to be related to asw2-a-eqiad, as shown by logs as well, but Faidon finds no further logs showing problems after 19:27
- 19:59 - about 7% packet loss seen from mw1312 to puppetmaster1001 (A6 to B6) is observed, giving further confirmation we're indeed having network problems
- 20:08 - Suspecting a fiber might be damaged, the VC link is disabled `<paravoid> asw2-a-eqiad: request virtual-chassis vc-port set interface member 6 vcp-255/1/0 disable`
- 20:08 - TCP retransmits and mcrouter TKOs & timeouts recover, packet loss stops **OUTAGE ENDS**
- 20:23 - logstash still choking on backlog & crashing due to UTF-8 parsing error [FATAL][logstash.runner] An unexpected error occurred! {:error=>#<ArgumentError: invalid byte sequence in UTF-8> (Suspected message example: https://phabricator.wikimedia.org/P8790)
- 21:00 - UTF-8 issue traced to kafka rsyslog-shipper input. Logstash collectors restarted with kafka rsyslog-shipper input disabled. Kafka-logging consumer lag begins to recover on non-rsyslog topics https://grafana.wikimedia.org/d/000000484/kafka-consumer-lag?orgId=1&from=1563904908998&to=1563929943574&var-datasource=eqiad%20prometheus%2Fops&var-cluster=logging-eqiad&var-topic=All&var-consumer_group=All
- 21:45 - UTF-8 issue traced to SlowTimer messages and a temporary fix to drop [message] =~ /^SlowTimer/ deployed
- 22:00 - temporary fix is active across eqiad/codfw logstash collectors, kafka rsyslog-shipper input re-enabled. Kafka consumer lag is recovering now on all topics
- 00:16 (next day) logstash has caught up with kafka backlog

Conclusions

While the overall impact of the incident was very limited, and that attests once again we can withstand issues with memcached - even the worst kind (the intermittent ones) - it showed we have a few badly-tuned bottlenecks - see what happened with logstash as an example. Also, our monitoring for network problems like this one is quite lacking.

The impact caused by memcache failures highlighted some bottlenecks in our infrastructure (most of them already known):

- The distribution of the shards in racks is not optimal, as it can be easily seen via Netbox. During the outage rack A6 was affected, containing 5/18 of the available shards (a big chunk of the whole cache layer). Moving shards from one rack to another is not very easy, especially due to the following bottleneck in the list. We have to accept this situation until we'll refresh the shards with new hardware (possibly with 10G NICs - spreading them in the racks in a more granular way).
- mcrouter is instructed to mark a shard with TKO (Temporary Knocked Down, hence not sending traffic to it)

after 10 timeouts of 1s. It then waits 4 seconds before starting to send health probes to verify if the shard is up (if the first is successful then the shard is removed from the TKO state). This means that timeouts to the 5 shards in rack A6 forced all the mcrouters on the MediaWiki app/api-appservers to mark them as down/TKO, without re-hashing the keys to other shards. This behavior is very different from what we were used to in the past with nutcracker, and described in T208934. In T208934 the SRE team is reviewing/evaluating a possible solution, namely adding failover support to mcrouter. This means adding a new memcache cluster in eqiad and one in codfw, to be used to accept traffic when one or more shards are not reachable.

What went well?

- We were able to identify the issue, and once we went to look into the network switches, we were able to solve the issue quickly
- Serving user's requests continued to work even during such a network problem, with the performance of ~30% of the memcached cluster severely degraded

What went poorly?

- We detected the issue from indirect alerts, and a second-level alert (logstash) is what paged us
- The log ingestion pipeline can't survive a large incoming volume of messages

Where did we get lucky?

- Logstash paging made more people come online and look at the issue than would have happened otherwise, in the absence of more precise alerting.

Links to relevant documentation

Network monitoring is the best source of information about such issues, but there is no runbook for "damaged fiber by activity in the datacenter", and it would be strange if it existed.

Incident_documentation/20190723-logstash describes the logstash outage that was closely related to this incident

Actionables

NOTE: Please add the #wikimedia-incident Phabricator project to these follow-up tasks and move them to the "follow-up/actionable" column.

- ~~Add basic metrics, alerts and observability for VCP port statistics, see T228824 - DONE~~
- Create a way to find out quickly, from the cli, which rack/row a machine is part of, from its ip/hostname. See the netbox backend for cumin already in the works.
- Attach a "rack" and "row" labels for all targets scraped from a given host, to make it easy to see rack-level and row-level aggregation of metrics (task: TODO)
- ~~Fix the faulty VC link T228823 - DONE~~
- Create a runbook entry to drop a particular class of messages from logstash's input (task: TODO)

Categories: Incident documentation in-reviews | Incident documentation