Page   Discussion

Read   View source   View history   Search Wikitech

Toolforge webservices are in the final stages of   migrating to the toolforge.org domain .
Please help us clean up older documentation referring to tools.wmflabs.org!

Incident documentation/20150413-LabsNFS

< Incident documentation

Summary

Labstore1001 became unresponsive to NFS because its block device subsystem was overloaded. A combination of factors caused a cascade failure resulting in complete resource starvation for NFS.

Timeline

```
17:30 Marc noticed NFS is slow during a fairly large copy, thinks the copy might
be the cause
17:34 Initial diagnostic on labstore1001 show NFS is starved out of disk
bandwidth
17:41 labstore1001 kernel stuck on kworker and bdflush processes, starved for IO
17:50 increase IO priority of NFS process hoping to improve interactive
performance
18:00 minor improvements visible, some NFS service restored, but not much
18:17 _joe_ joins investigation
18:20 one of the shelves is rebuilding raid6, suspected of being an issue but
reducing bandwidth does not visibly help
18:29 stoping NFS and unmounting filesystems to cold start labstore1001
18:33 cold start of labstore1001, go into MD800 BIOS to check diagnostics
18:36 No failures in the hardware reported, proceeding with boot
18:44 NFS back up,
18:53 NFS spotty at best, very high iowait still noticable with disk usage
pegged
19:02 bblack joins investigation
19:33 Another cold start to do the bootstrap manually, trying to isolate the
problem component
19:38 attempt to let the raid resync proceed, would take 20h at current (overly
slow) rate
20:05 mark suggest tuning stripe_cache_size to increase rebuild speed.
Increases efficiency tenfold.
20:08 noted how raid6 is bound to a single CPU in labstore1001's older kernel,
no further improvement in raid6 speed possible
20:12 reduce rebuild speed to leave some IO bandwidth, restart NFS
20:22 NFS returns to reasonable working order, with some intermitent
sluggishness
20:47 Most things return to working order, while Coren and Yuvi restore some
services that did not survive the outage
21:12 All services back to normal, but iowait remains high
01:34 iowait on labstore1001 returns to normal patterns.  Cause unknown as
rebuild still in progress.
```

Conclusions

There seems to be no single, isolable cause to the outage. Rather, a combination of factors seem to have resulted in the demand on disk bandwidth to exceed the capacity of the system to the point where cascading failure was reached. The kernel stripe_cache_size being set to the default (too small) kernel value amplified the drain on resources caused by the raid resync to the point where a buffer flush initiated by the kernel ended up

starving all processes out of disk bandwidth. In addition, the older kernel (from Precise) has a bottleneck on raid6 checksum calculation being single-CPU bound that aggravated matters.

Actionables

- Move to a more modern kernel as swiftly as practical https://phabricator.wikimedia.org/T94609 🔗
- make certain the stripe_cache_size setting is puppetized and applied at boot https://phabricator.wikimedia.org/T96045 🔗
- Formulate plans for getting off of raid6 for labs NFS storage. https://phabricator.wikimedia.org/T96063 🔗
- Formulate plans for reducing unnecessary NFS I/O by pushing projects to use local storage for heavy i/o traffic. https://phabricator.wikimedia.org/T96065 🔗

Category: Incident documentation