



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/20120916-GeoDataTemplate

[< Incident documentation](#)

Outage Summary

Wikimedia sites experienced 3 episodes of intermittent performance lags and brief unavailability on 16th & 17th August 2012. The first two incidents occurred on 16th at 13:22 UTC and at 15:30 UTC. The third incident was on 17th September at 10:40 UTC.

- **Duration:** The durations of those incidents were about 1.5 hours, 1.5 hours and 3.5 hours respectively
- **Impact:** During those episodes, editors on our Wikimedia sites lost their sessions intermittently and briefly. Some readers did experience response lag that caused pages to time out thus not rendered.
- **Cause:** There were load spikes on the Apache servers, each time it caused intensive page swapping in the memory and that overload caused the Apache servers to reboot themselves during prolonged spikes. Further analysis revealed the main cause of this high memory usage to be geographical data ("données") templates on the French Wikipedia, and to a lesser extent, the same data templates copied to other wikis for use on articles about places in Europe and in zhwiki
 - The zhwiki job queue has been growing quicker than the job runners can run them.
 - During investigation there are 380303 (zhwiki) jobs in the queue, with 808 distinct job_titles (There were jobs duplication 1K times). Duplicate removal has been broken for a long time (ever since we had jobs with start/end parameters).
 - Template DOM cache issue resulted in expanding memory needs, thus the frequent memory swaps and Apache failures.
- **Resolution:** Longer term fixes are being investigated and would take some time to implement. Some temporary measures were taken, such as:
 - contact the few Wikipedians involved to stop doing what they are doing, and to remove the template invocations which have already been introduced. Antoine Musso has raised the issue on the French Wikipedia's "Bistro" and some of the

worst cases have already been fixed.

- taking out the site notice gadget on zhwiki that was creating many non-cache objects
- Reducing the MaxRequestsPerChild from 300 to 4 in apache.conf
- limiting the number of large template DOM caches from sending servers into swap, by throwing an exception when more than some number of DOM elements are parsed.
- Tim introduced a new preprocessor complexity limit called the "preprocessor generated node count", which is incremented by about 6 for each #switch case. When the limit is exceeded, an exception is thrown, preventing the page from being saved or viewed.

Detail

Below are the reports from Mark, Tim and Asher, following the incidents.

From Mark (following the 1st incident):

We've just had a site outage, starting around 13:22 UTC when the MediaWiki/Apache cluster became unhappy, and as a result of that, other clusters such as the Text Squids as well. Leslie, Peter and I started looking at it when we got woken up by the texts. The Apaches were running into swap death, causing general unresponsiveness across the board, including the Memcached instances on those boxes and retries/refetches from the Squids, and thus amplifying the problem further. All we really knew was that Apache processes were rapidly growing in memory size, probably PHP. Debugging of the problem was hindered by the fact that these boxes would become completely unresponsive quickly.

Some measures against the symptoms were taken such as restarting Apache and power cycling unresponsive boxes, but this was of course not helping to fix the root of the problem. As often, the problem would sometimes disappear eventually with the site becoming responsive again, but it never lasted more than

[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](#)

10 minutes.

Eventually we found through tcpdump/ngrep on individual Apaches an unusually large amount of requests for:

```
/w/index.php?title=Template%3AAdvancedSiteNotices%2Fajax&variant=zh-tw&action=render
```

on zh.wikipedia.org. This seems to be some site notice Gadget enabled on zhwiki, and this page was served as uncacheable. Andrew Garrett then disabled this Gadget. Although we can't definitively say this was the issue, the sites have been stable since.

We should also really look into getting some better instrumentation for finding problematic requests I think. Currently it's rather hard to figure out what a problematic Apache/Mediawiki instance is really doing, or to find unusual/new request patterns.

From Tim:

It was probably the template DOM cache, like the owiki problem. I've pushed a change to Gerrit which will throw an exception if more than a certain number of DOM elements are created:

```
https://gerrit.wikimedia.org/r/23921
https://gerrit.wikimedia.org/r/23922
https://gerrit.wikimedia.org/r/23923
```

It's configured with \$wgMaxGeneratedPPNodeCount. The memory usage is probably somewhere in the vicinity of 100 to 300 bytes per element, so the default of 1 million will give a maximum of ~100-300MB memory usage. After deployment, the exception log (fluorine:/a/mw-log/exception.log) needs to be watched closely to make sure the change was not too disruptive to existing articles.

If the problem happens again, deploying this change would probably be a good idea.

From: Mark following Sept 17 incident:

So we just got woken up by another outage. The Apaches were not nearly out of memory this time, so it wasn't this problem again. PyBal on lvs4 seemed very unhappy with all apaches, depooling and repooling them at random. I noticed that the Parser cache has been at a consistent 1000ms latency for the past day:

```
https://graphite.wikimedia.org/render/?
title=Parser%20Cache%20Hit%20Latency%20(ms)%20-
1week&vtitle=time%20in%20ms&from=-
1week&width=1024&height=500&until=now&areaMode=none&hideLegend=false&lineWidth=2&
lineMode=connected&target=color(cactiStyle(alias(ParserCache.get.tavg,%22Avg%22)),%
22green%22)&target=color(cactiStyle(alias(ParserCache.get.tp99,%2299%25%22)),%22red%22)
```

I cleared profiling data. Then noticed that Memcached gets were at roughly 50% of real time, with 2 down Memcached instances in the top profiling report, so I power cycled those. Actually I power cycled srv250 but instead srv249 went down. (Chris/Rob, please fix!) For some reason "mwscript mctest.php" was not complaining about those two memcached instances, which made me search elsewhere for a while first.

After the two down memcached instances came back up, everything stabilized again, and PC hit latency was down to normal.

From Asher after further investigation:

The zhwiki job queue has been growing quicker than the job runners can run them. I've manually deleted around 150k duplicates (our deduplication is completely broken due to the inclusion of mysql master positions in job_params which is used in full for the de-dupe attempt) but the trend still doesn't look good. The jobs are all of the refreshLinks2 variety and most are from template edits. At the moment there are 380303 jobs in the queue, with 808 distinct job_titles.

I traced the full run of the following job "2012-09-18 06:25:51 refreshLinks2 Template:Country_data_JAM table=templatelinks start=818486 end=854177 masterPos=db54-bin.000375/120945239 t=41901 good"

This one job took ~42 seconds to run (from 06:25:09.631586 to 06:25:51.544876)

It required 957 template inclusions (a few: [Template:Country data Finland](#) [Template:Country data Fiji](#) [Template:Country data Falkland Islands](#) [Template:Country data Federated States of Micronesia](#))

4439 SELECT mysql queries

7463 memcached gets

0 (zero) INSERTIREPLACEIUPDATEIDELETE mysql queries (including to parsercache)

0 (zero) memcached writes (based on examining all sendto calls to all 79 open sockets to memcached)

Category: [Incident documentation](#)

This page was last edited on 19 September 2016, at 21:10.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. See [Terms of Use](#) for details.

[Privacy policy](#) [About](#)

[Disclaimers](#) [Code of Conduct](#) [Developers](#) [Statistics](#) [Cookie statement](#) [Mobile view](#)

[Wikitech](#)

