CUNY Academic Commons - Bug #12436

Nightly system downtime

2020-02-18 11:21 AM - Boone Gorges

Status:	Assigned	**Start date:**	2020-02-18
Priority name:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category name:		**Estimated time:**	0.00 hour
Target version:	Not tracked		

Description

I'm opening this ticket to track recent outages on the Commons. A number of us receive automated notices when the database server is forced to reboot. Beginning roughly one month ago, we started getting these notices several times per week. These notifications indicate when the DB server is forced to reboot, so they indicate at best the end of a problematic period. I receive additional notifications when requests begin to take an inordinately long time, which serves as an indicator of when the incidents begin.

I've been keeping track of specifics over the last few weeks, and I've discerned the following patterns:

- The most common downtime is just after 05:00 UTC (midnight EST, UTC-5), with my incident reports rolling in sometimes around 12:03am and sometimes around 12:07am.
- Occasionally, the incidents have begun an hour or two earlier, shortly after 03:00 or 04:00 UTC.
- Incidents seem always to begin several minutes after the hour
- Reboots usually take place between 3 and 6 minutes after the beginning of the incident
- On some occasions, the reboots don't seem to fix the underlying issue, and another cycle of slow requests + db reboots immediately follows.
- Sometimes this'll happen a few nights in a row, while sometimes the site will go a few days without any notifications.

All of this strongly suggests that the problem is with an automated cron task, specifically one that takes place around midnight. I've begun to do an analysis of the tasks scheduled for around this time, cross-referencing with the logs. Ideally, we'd be able to narrow down the culprit by identifying the last cron task that begins just before the outages. This is not possible, for a couple reasons: First, the performance issues may only kick in a minute or two after the task begins running (as the system's resources are gradually used up). Second, the Cavalcade logs don't natively keep track of when a task begins running, but only when it finishes (see https://github.com/humanmade/Cavalcade-Runner/blob/master/inc/class-runner.php#L377).

So the best we can do is to make some educated guesses. I'll follow up in a comment with initial thoughts.

Related issues:

Related to CUNY Academic Commons - Feature #12440: Upgrade Cavalcade	**Resolved**	**2020-02-18**

History

#1 - 2020-02-18 12:02 PM - Boone Gorges

Looking over tasks that finished in the first 15 minutes after 0500 UTC last night (when we know an incident occurred) we can narrow down the possibilities somewhat. Here is a unique list of hooks fired during this time period:

```
| akismet_scheduled_delete                    |
| blc_cron_check_links                        |
| bpges_health_check                          |
| btcnew_connect                              |
| delete_expired_transients                   |
| do_pings                                    |
| embedly_revalidate_account                  |
| jetpack_sync_cron                           |
| jetpack_sync_full_cron                      |
| jetpack_v2_heartbeat                        |
| newsletters_countrieshook                   |
| ngg_delete_expired_transients               |
| pj_wpdb_sessions_gc                         |
| pull_feed_in                                |
| take_feed_out                               |
| tribe_aggregator_cron                       |
| tribe_events_pro_process_recurring_events |
| wp_privacy_delete_old_export_files          |
| wp_rss_multi_event                          |
| wp_rss_multi_event_delete_custom_posts      |
| wp_scheduled_auto_draft_delete            |
| wp_scheduled_delete                       |
| wpbdp_hourly_events                       |
| wpbdp_listings_expiration_check           |
| wpml_autoresponders                       |
| wpml_captchacleanup                       |
| wpml_cronhook                             |
| wptouch_cron_backup_settings              |
| wsd_cleanup_live_traffic
```

Some of these can reasonably be eliminated for various reasons: 1. They are known to run fairly frequently and they don't trigger problems at any other time of day, so it's reasonable to think they're not the root cause here; 2. The tasks they perform are known to be modest (like deleting expired transients) and so are likely unrelated. In contrast, the first place to look is those hooks that trigger processes known to be resource-intensive, specifically in a way that would hold open connections. This brings to mind plugins that pull in RSS feeds: not only do they make (locking) external requests, but they make lots of them in a row.

1. blc_cron_check_links comes from broken-link-checker
2. pull_feed_in is from pressforward
3. wp_rss_multi_event is from wp-rss-multi-importer

I suspect that one of these is at the root of the issue. Unfortunately, it's not possible to determine this by looking at the logs (for reasons discussed above) or by looking at the code (because the specific RSS feeds on a given site might trigger a problem that wouldn't exist on other sites running the plugin).

As an initial shot-in-the-dark test, I'm going to disable the cron jobs for each of these plugins in turn, to see if the problems go away. I'll start with the first one, broken-link-checker, on a hunch. It is active on 86 sites across the Commons. I'll short-circuit its cron jobs and check back tomorrow. If we experience the same outages, we can eliminate this one from contention and move onto the next. If there are no outages, we will have to wait a few more days to feel confident about the diagnosis.

#2 - 2020-02-18 12:06 PM - Boone Gorges

broken-link-checker cron job disabled in https://github.com/cuny-academic-commons/cac/commit/f326b1850cbe1ad586fa823667247413176eba5a

#3 - 2020-02-18 03:08 PM - Raymond Hoh

Semi-related, Cavalcade just released a new version today:
https://github.com/humanmade/Cavalcade/releases/tag/2.0.0

It includes a major change to how Cavalcade stores the cron data from WP. We might want to wait a few maintenance releases because I'm quite sure that there will be bugs.

#4 - 2020-02-18 03:30 PM - Boone Gorges

- Related to Feature #12440: Upgrade Cavalcade added

#5 - 2020-02-18 03:31 PM - Boone Gorges

Thanks, Ray. I've opened #12440 to track this.

#6 - 2020-02-19 10:09 AM - Boone Gorges

We had more reboots last night, so I think we can rule out broken-link-checker for now. I've reenabled its schedule task in
https://github.com/cuny-academic-commons/cac/commit/5f2cc92be0a97bf84a4e701603b90f90f3bb8194.

I analyzed the cron logs corresponding to last night's outage times. wp-multi-rss-importer's cron job is not present in all of them, but pressforward's is, which leads me to suspect the latter. I've disabled its 'pull_feed_in' job in
https://github.com/cuny-academic-commons/cac/commit/1677c15f3c0df45a460dfa59fcc68aff01b9afe7 and will check in tomorrow.

#7 - 2020-02-20 01:42 PM - Boone Gorges

Looks like PressForward is not to blame. My next guess is that it has something to do with Jetpack's sync module: it runs many jobs, and of the requests that were especially long-running, a number were Jetpack-related syncs. I'm going to disable it temporarily to see if it's the culprit.

- pressforward cron jobs reenabled in https://github.com/cuny-academic-commons/cac/commit/9d80def316db4262eac1791dff79e61837bdcbf3
- jetpack sync cron jobs disabled in https://github.com/cuny-academic-commons/cac/commit/2b0dbe5a65bcb6cf116500f6f51dd28f424b2c67

#8 - 2020-02-21 10:35 AM - Boone Gorges

Another outage last night, so Jetpack sync is off the hook. Reenabled in
https://github.com/cuny-academic-commons/cac/commit/c1de6beb7941cd4a6e0d227e8ee660abc63c7c86.

Moving on at this point to Akismet, whose cron jobs are among the most frequently run on the system. It's possible that there's one or two sites with an enormous amount of spam, such that the scheduled delete causes the system to hang. Disabled for the time being in
https://github.com/cuny-academic-commons/cac/commit/99ce2b3439f314a7a861487b082a6531f58671cc.

#9 - 2020-02-22 09:53 AM - Boone Gorges

Reenabled akismet in https://github.com/cuny-academic-commons/cac/commit/3bd974c8e8f135e04bed35dd6a3bd494058b6e1c

Disabled newsletters-lite cron jobs in https://github.com/cuny-academic-commons/cac/commit/751d7997750645ca9181568043161ec3cbb5acd4

#10 - 2020-02-24 11:16 AM - Boone Gorges

Restored newsletters-lite cron tasks in https://github.com/cuny-academic-commons/cac/commit/89dd3d4741e17cb0451fe48dd8303f1ab389ad15

I'm starting to run out of obvious culprits in the cron queue. My original assumption had been that, since the outages were always at the same time, that it's a WP scheduled task causing the problem. But I'm becoming less sure about this - it could be that there's a **server** scheduled task triggering the problem. As a test, I've put some code in place that will disable all cron jobs scheduled to run between 04:59 and 05:15 UTC on 2020-02-25 (one minute before midnight NYC time, and for sixteen minutes afterward). If we still see the outage, we know that WP cron jobs are not the root cause.

#11 - 2020-02-25 10:37 AM - Boone Gorges

There were no outages last night, which suggests that disabling cron jobs during this period did have the intended effect. (Could also be a fluke, but I have limited info to go on.)

As a new strategy, I'm going to try bulk-disabling certain classes of cron jobs. The first pass is that I'll block all cron hooks run between 04:59 and 05:15 UTC that start with letters a-m. If we have outages, I'll switch to the last half of the alphabet. And I'll attempt to narrow things down in this fashion. https://github.com/cuny-academic-commons/cac/commit/0143ad9801b2205463e4dedf970f8e49064470e6

#12 - 2020-02-27 11:42 AM - Boone Gorges

We've had two nights without midnight outages after the change above, so I think it's reasonable to assume that the problematic hook(s) start with a-m. I'm going to narrow it down by allowing h-m (continuing to block a-g) and we'll see what happens. https://github.com/cuny-academic-commons/cac/commit/972388e626ec4a00617b4f84174319add41c11bc

#13 - 2020-02-28 02:51 PM - Raymond Hoh

Boone, came across a fatal error from the last Cavalcade debugging: https://github.com/cuny-academic-commons/cac/commit/4807f5530270e1d10f5277670e2435591229f2dc

We'll probably have to let Cavalcade run for another day to see if the nightly downtime happens again.

#14 - 2020-02-28 02:52 PM - Boone Gorges

Whoops. Thanks for catching this, Ray.

#15 - 2020-03-10 11:06 AM - Boone Gorges

- Status changed from New to Abandoned

- Target version set to Not tracked

An update. I continued to pare down the blacklisted Cavalcade hooks over the course of about a week. For the past two or three days, only those with beginning with the letter A have been blocked. During this time there have been no reboots in the night. However, my logs show that this blacklist only blocks the akismet_scheduled_delete hook, which has only run around 12:50am - not near midnight. (And since DST began, it's been even later.) So I think that we've gone too far down the garden path.

Given that the downtime is not currently recurring, I'm setting this ticket to one side for the time being. I'll reopen and back up investigations a bit if we see it pop back up.

#16 - 2020-07-20 09:56 AM - Boone Gorges

- Status changed from Abandoned to Assigned

Nearly identical behavior has begun to crop up over the last week. Nightly service disruptions, beginning around 12:03 or 12:04am EDT, usually resolving with a DB reboot.

One thing to note (not sure if it's relevant) is that it continues to happen just after midnight Eastern time, but now we are in DST, which makes it 0400 UTC. This indicates that the problem is tied to a scheduled task that's sensitive to local time, not to UTC.

As a starting point, I am once again disabling scheduled tasks beginning with a-m between 0359 and 0415. Based on this, we'll whittle down the exceptions a bit.

#17 - 2020-07-23 08:07 AM - Boone Gorges

Continuing to see the problems with a-m disabled, so I'm going to try n-z instead. https://github.com/cuny-academic-commons/cac/commit/540f6107492621a76a2058da2a9aed51b733400b

#18 - 2020-07-27 12:10 PM - Matt Gold

Thanks so much for your continued work on this, Boone -- Luke and I noticed the nightly messages again and I'm so glad you are on the case, difficult as it is. Great detective-level insight on the timing!

#19 - 2020-08-01 09:30 AM - Boone Gorges

After a few more unsuccessful modifications to my debug script, I got in touch with Lihua. He replied that the IT team is aware of broader issues causing downtime across their systems around this time. As such, he recommended that we not sink any more time into debugging this at the moment.

I've disabled our debugging tools in https://github.com/cuny-academic-commons/cac/commit/d9e6005bc3a6778fd5bb8f597a26c85099bc5fe6