

CUNY Academic Commons - Feature #12440

Upgrade Cavalcade

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

Status:	Resolved	Start date:	2020-02-18
Priority name:	Normal	Due date:	
Assignee:	Raymond Hoh	% Done:	0%
Category name:	Cavalcade	Estimated time:	0.00 hour
Target version:	1.18.8		
Description			
See https://redmine.gc.cuny.edu/issues/12436#note-3 , https://github.com/humanmade/Cavalcade/releases/tag/2.0.0			
Related issues:			
Related to CUNY Academic Commons - Bug #12436: Nightly system downtime		Assigned	2020-02-18
Related to CUNY Academic Commons - Bug #14199: Email replies delayed		Resolved	2021-03-18
Related to CUNY Academic Commons - Bug #14276: Increase number of workers in ...		Deferred	2021-04-02

History

#1 - 2020-02-18 03:30 PM - Boone Gorges

- Related to Bug #12436: Nightly system downtime added

#2 - 2020-07-14 12:20 PM - Boone Gorges

- Target version changed from 1.17.0 to 1.18.0

#3 - 2020-09-23 10:35 AM - Boone Gorges

- Category name set to Cavalcade

#4 - 2020-09-23 10:59 AM - Boone Gorges

- Assignee changed from Boone Gorges to Raymond Hoh

Ray, could you please take the lead on this? I think it'll look something like the following:

- Cavalcade is made up of two parts: <https://github.com/humanmade/Cavalcade-Runner> (running as a systemd script on the webserver) and <https://github.com/humanmade/Cavalcade> (running as a mu-plugin in the Commons codebase). Compare the latest versions of each with the versions we're running, and make note of those changes that might require intervention. Eg, I think that we might have to run a database schema upgrade; see <https://github.com/humanmade/Cavalcade/commit/2ce546f2d075de3eeebf5095d752ae471b7477fa>
- Get the system running locally and run some basic tests to ensure that jobs are properly queued (this is the mu-plugin) and then run (the daemon) using the updated software
- Reach out to Lihua to get the new daemon running on LDV2 and run tests there

I know I did most of this the first time around, but I thought it would be helpful for someone other than me to have a sense of how it works, and I thought this would be an OK way for you to get that knowledge. But let me know if you've got technical limitations from taking on the task :)

#5 - 2020-09-28 05:50 PM - Raymond Hoh

Get the system running locally and run some basic tests to ensure that jobs are properly queued (this is the mu-plugin) and then run (the daemon) using the updated softwa

Hi Boone, my Commons instance is on a WAMP stack, which doesn't support `pcntl` (`pcntl` is required for the runner). So I cannot test the Commons locally with Cavalcade at the moment, but I could test on the development server. (I should probably spend some time to migrate my Commons WAMP stack to using a LAMP stack within Windows at some point in the future!)

If you're okay with me testing on the development server, let me know.

#6 - 2020-09-29 11:12 AM - Boone Gorges

Fine with me to test on the dev site. Thank you, Ray! And if it turns out that there's some sort of testing that needs to be done locally, just bump this ticket back over to me.

#7 - 2020-12-08 11:31 AM - Boone Gorges

- Target version changed from 1.18.0 to 1.19.0

#8 - 2021-03-19 11:26 AM - Boone Gorges

Coping over some comments from [#14199](#):

I tried removing RBE's bp_rbe_schedule event and re-added it, but after the scheduled time, the task is still marked as waiting with the nextron timestamp being the same as the start timestamp.

It just occurred to me that daylight savings time happened over the weekend, which might be related to this problem. I believe this is a reschedule bug with the Cavalcade Runner, which is described here:

<https://github.com/humanmade/Cavalcade/issues/74#issuecomment-549148047>

And addressed with this PR: <https://github.com/humanmade/Cavalcade-Runner/pull/64/files>

Also related: <https://github.com/humanmade/Cavalcade-Runner/issues/51>

Ah, good find, Ray. Sounds like the DST issue is probably the one that we're experiencing.

Lihua would have to be responsible for upgrading our instance of cavalcade-runner. My feeling is that we should probably upgrade Cavalcade (mu-plugins) at the same time. I've looked briefly at the changelogs and I don't see any obvious reason why we shouldn't be able to do this.

Could you also have a look and let me know what you think? <https://github.com/humanmade/Cavalcade-Runner/commits/master>

<https://github.com/humanmade/Cavalcade/commits/master>

It looks like Cavalcade has made a number of database query changes since we initially installed. They introduced an upgrade routine <https://github.com/humanmade/Cavalcade/blob/master/inc/upgrade/namespace.php> but it depends on the cavalcade_db_version database option, which we don't have - I think that our installation predates this. I think this will be fine because the (int) get_site_option() call will result in a 0.

The prudent path forward is probably to do an upgrade of both packages on LDV2 and do some brief testing there. I'll get in touch with Lihua to start this process. Let's move this discussion over to [#12440](#).

Ray, did you ever reach out to Lihua to do this upgrade, as we discussed above? I can't remember :)

#9 - 2021-03-19 02:36 PM - Raymond Hoh

- Related to Bug #14199: Email replies delayed added

#10 - 2021-03-19 02:45 PM - Raymond Hoh

Ray, did you ever reach out to Lihua to do this upgrade, as we discussed above? I can't remember :)

I did not get to this I'm afraid. I can reach out to him now so we can get started on the Cavalcade testing on the development server or you can do so as well.

#11 - 2021-03-19 03:01 PM - Boone Gorges

Would you mind doing it? Thanks, Ray!

#12 - 2021-03-22 07:46 PM - Raymond Hoh

I've updated Cavalcade to v2.0.2 in a separate branch -

<https://github.com/cuny-academic-commons/cac/commit/9cd921148307d8f14cff94e07ba6c894684e5225>. And deployed it on the dev server. I also ran the Cavalcade database upgrade routine with wp cavalcade upgrade and tested with adding a sample cron job and the job ran successfully.

I've also just emailed Lihua about upgrading the Cavalcade Runner. I'm not sure if we can upgrade just the runner on the dev server before deploying on production, but will wait to see what Lihua says.

#13 - 2021-03-23 04:47 PM - Raymond Hoh

Thanks to Lihua, I tested the updated Cavalcade runner on the dev site and it works as expected. I scheduled one single cron task and one recurring cron task to see if they would run and both tasks ran successfully.

Boone, any other things we should test for before deploying on production?

#14 - 2021-03-23 05:34 PM - Boone Gorges

I think you can deploy anytime. Thanks, Ray!

#15 - 2021-03-24 12:22 PM - Raymond Hoh

Oops! Posted the following reply in the [wrong ticket](#) :

I updated the plugin portion of Cavalcade on production [last night](#). We had a few additional indexes on our wp_cavalcade_jobs table that are not part of Cavalcade's database schema, so I removed them.

I just pinged Lihua so the runner portion will be updated sometime later today.

#16 - 2021-03-25 05:07 PM - Raymond Hoh

Just to update, Lihua has updated the runner on lw2b for now.

One thing I just recognized is we have a huge job queue on production. Running the following MySQL query -- select count(id) from wp_cavalcade_jobs where nextrun < NOW() and status = "waiting" -- as of right now returns ~60900 items.

I just ran the same query after a minute and that number actually went up (!), which means more jobs are being added than the number of workers can get through.

This Cavalcade Runner issue is applicable - <https://github.com/humanmade/Cavalcade-Runner/issues/51>.

How the runner currently reschedules tasks is just by tacking on the interval to the last saved nextrun time in the DB: <https://github.com/humanmade/Cavalcade-Runner/blob/0dfb42d505e9cd870a11366c49ee680d327c961a/inc/class-job.php#L87-L89>. This means that if there is a huge backlog where the nextrun time is behind the current time by a lot, this rescheduling will take forever to catch up (if at all). The DST switchover exposes this problem.

It looks like the backlog is behind by three days. I would probably recommend rebuilding the wp_cavalcade_jobs table as recommended here - <https://github.com/humanmade/Cavalcade/issues/74#issuecomment-435741445>.

#17 - 2021-03-25 05:46 PM - Boone Gorges

Yes, it seems to me that it will probably never catch up.

It's definitely fine to trim and/or truncate the log table.

As for the jobs table, here's what the GitHub commenter says:

(e.g. I only kept the one-off events; the rest could reschedule themselves)

Is that actually true in most cases? If so, I think it's OK to move forward with this strategy. Actually, we can probably look at this on a case-by-case basis. Here are the top ten most frequent hooks in the jobs table, which account for the vast majority of entries:

```
mysql> select hook, count(*) from wp_cavalcade_jobs group by hook order by count(*) desc limit 10;
+-----+-----+
| hook                                     | count(*) |
+-----+-----+
| wp_site_health_scheduled_check         | 16462    |
| enable_jquery_migrate_helper_notification | 13435    |
| wp_privacy_delete_old_export_files     | 12441    |
| wp_scheduled_auto_draft_delete        | 12126    |
| wp_scheduled_delete                   | 7350     |
| delete_expired_transients              | 7287     |
| jetpack_clean_nonces                   | 4874     |
| jetpack_v2_heartbeat                   | 3647     |
| akismet_scheduled_delete              | 2610     |
| et_core_page_resource_auto_clear       | 2196     |
+-----+-----+
```

Of those:

- wp_site_health_scheduled_check - will reschedule itself
- enable_jquery_migrate_helper_notification - will reschedule itself, though I think we can probably actually block these altogether since we don't allow site admins to see the notice anyway. See <https://github.com/cuny-academic-commons/cac/blob/1.18.x/wp-content/mu-plugins/cavalcade.php>
- wp_privacy_delete_old_export_files - will reschedule itself
- wp_scheduled_auto_draft_delete - will reschedule itself next time the author goes to write something (and if it that doesn't happen soon, it doesn't matter)
- wp_scheduled_delete - will reschedule next time the admin loads (again, doesn't matter if it doesn't happen)
- delete_expired_transients - will reschedule itself
- jetpack_clean_nonces - will reschedule itself
- jetpack_v2_heartbeat - will reschedule itself
- akismet_scheduled_delete - will reschedule itself on the next comment
- et_core_page_resource_auto_clear - will reschedule itself

Based on this, I think we can very safely delete at least those jobs belonging to the top-ten offenders. Ray, can you confirm this logic before I pull the

trigger on it?

#18 - 2021-03-25 05:58 PM - Raymond Hoh

(e.g. I only kept the one-off events; the rest could reschedule themselves)

Is that actually true in most cases?

I believe so. In most cases, any recurring job would require the plugin to check if it is already scheduled before scheduling it again.

However, I don't mind going conservative by eliminating the top offenders. +1 from me!

#19 - 2021-03-25 06:18 PM - Boone Gorges

Thanks! I have deleted those in this top ten. They are rapidly rescheduling themselves, and the 'waiting' count is back up to 4306. But many of these are for weekly or daily events, so let's see if things start to hit an equilibrium after a while.

#20 - 2021-03-25 07:11 PM - Raymond Hoh

Since we have a backlog of three days, I think we will also want to remove all "waiting" items that have a short interval because they will take a long time to catch up.

Perhaps these ones with an interval of less than one hour?

```
select hook,count(*) from wp_cavalcade_jobs where nextrun < NOW() and status = "waiting" and `interval` < 3600
group by hook order by count(*) desc'
```

hook	count (*)
jetpack_sync_full_cron	119
jetpack_sync_cron	112
blc_cron_check_links	34
pull_feed_in	9
action_scheduler_run_queue	9
wp_gf_feed_processor_cron	3
elementor_9669_elementor_updater_cron	1
elementor_13252_elementor_updater_cron	1
useyourdrive_synchronize_cache	1
wp_gf_upgrader_cron	1
bpges_health_check	1

The only one that needs rescheduling is bpges_health_check.

#21 - 2021-03-25 08:20 PM - Boone Gorges

Yeah, good idea. Could you go ahead and make the change? Make sure to flush the cache afterward.

#22 - 2021-03-26 02:00 AM - Raymond Hoh

I removed those jobs (and then some!) and flushed the cache with Cavalcade's built-in method:

<https://github.com/humanmade/Cavalcade/blob/e8b1e9a08d242559f82fd9a0eb59e3ea2ef968f0/inc/class-job.php#L383>

It took awhile, but it looks like the jobs have finally caught up!

#23 - 2021-03-26 09:59 AM - Boone Gorges

- Target version changed from 1.19.0 to 1.18.8

Amazing, thanks!

Ray, I'll let you do the honors of closing this ticket once you're satisfied.

#24 - 2021-03-29 03:14 PM - Raymond Hoh

- Status changed from New to Resolved

Lihua just upgraded the Cavalcade runner on all production nodes so going to close this one out!

#25 - 2021-04-02 05:36 PM - Raymond Hoh

- Related to Bug #14276: Increase number of workers in Cavalcade added