

CUNY Academic Commons - Bug #5268

Long-time to post to multiple groups

2016-02-26 12:19 PM - Luke Waltzer

Status:	Assigned	Start date:	2016-02-26
Priority name:	Normal	Due date:	
Assignee:	Daniel Jones	% Done:	0%
Category name:	Group Forums	Estimated time:	0.00 hour
Target version:	Future release		
Description			
Hi All-- put a forum post up yesterday to multiple groups -- https://commons.gc.cuny.edu/groups/teaching-and-learning-center/forum/topic/teaching-with-open-educational-resources-march-2-1-pm -- and the post took more than a minute to go through. Not sure if this is expected behavior but wanted to flag it.			

History

#1 - 2016-02-26 12:21 PM - Matt Gold

- Status changed from New to Assigned
- Assignee set to Boone Gorges

Thanks for reporting, Luke

#2 - 2016-03-01 10:31 PM - Boone Gorges

- Category name set to Group Forums
- Target version set to Future release

I've spent some time profiling the process. I can confirm that it's fairly slow, but I'm afraid there's no silver bullet solution.

One relatively costly part of the posting process is that the wordpress-mu-sitewide-tags plugin runs a sort of cleanup routine every time a post anywhere on the network transitions its post_status. From what I can see, this routine is completely broken - the query it attempts to run is quite resource-intensive, and can never successfully match a result. I've disabled it in <https://github.com/cuny-academic-commons/cac/commit/746b40c744b1a464af2cd4f1d7af5b364d178599>, which should go some way toward speeding up forum posting (and every other kind of post creation on the entire network).

Aside from this, the underlying problem is that bbPress is pretty slow. When creating a topic, bbPress writes to the postmeta table dozens of times; when spawning lots of topics in a single pageload, the overhead increases linearly. There's not a lot that we can do for this, except to explore the possibility of offloading the crossposting to a background routine. I'm going to assign this ticket to Dan for a bit of thought and research about whether this is feasible (given the current plugin's reliance on global state, etc).

#3 - 2016-04-17 02:04 PM - Daniel Jones

Came upon this by accident when I was looking at future release issues! Boone I just wanted to see if you meant to hold off on assigning this to me or if I should start looking into it. I've also noticed that it's really slow.

#4 - 2016-04-17 02:20 PM - Daniel Jones

This looks like a promising and lightweight library. I think it works by offloading routines to wp_remote_post() calls:
<https://github.com/A5hleyRich/wp-background-processing>

Is wp_remote_post() really non-blocking?

#5 - 2016-04-21 12:33 AM - Boone Gorges

- Assignee changed from Boone Gorges to Daniel Jones

Dan - Please do take a look at implementing background processing for this. If you can get it to work, there are a half dozen other places in the Commons codebase where I think we could use a similar technique, to great effect :)

You may run into issues with data that's only available at runtime. Use whatever technique you deem necessary to store it for offloaded processing. Please post to the ticket here (I've added Ray as a watcher too) if you have any questions or want to bounce any ideas off of us.

#6 - 2016-06-15 02:58 PM - Daniel Jones

So the problem I'm running into here is the need to pass the ID of the newly generated post back to the main thread, after processing in the

background. It doesn't look like there's an easy way to do it, and I think it'd defeat the purpose of doing it in a non-blocking way anyway.

One thing that might be easier and might still make a difference could be to just do the email-sending in a background process. I think that that takes a long time, and doing it in the background might really make a difference. Should I give that a try?

#7 - 2016-06-15 10:09 PM - Boone Gorges

Sending emails in batches is going to run into the same problem as posting forum topics in batches: you need to store a bunch of data to be run in separate processes.

Ray has done a bunch of thinking about this with respect to the BuddyPress Group Email Subscription plugin. Maybe he can weigh in with some thoughts?

#8 - 2016-06-22 09:53 PM - Raymond Hoh

So the problem I'm running into here is the need to pass the ID of the newly generated post back to the main thread, after processing in the background. It doesn't look like there's an easy way to do it, and I think it'd defeat the purpose of doing it in a non-blocking way anyway.

Dan, do you have any example code somewhere?

I would stick with TechCrunch's WP_Async_Task library:

<https://github.com/techcrunch/wp-async-task>

TechCrunch's library is easier to understand and quite simple to set up once you understand their approach.

What you would do is extend the WP_Async_Task class and tailor it to the 'bbp_new_topic_post_extras' action because that is the hook you're using to do the multiple post email sending (bpmfp_create_duplicate_topics()). WP_Async_Task::prepare_data() is where you would pass the topic ID from the 'bbp_new_topic_post_extras' hook.

Hopefully that is enough to go on.

For more examples, see how I'm using WP_Async_Task in the experimental Group Email Subscription branch:

<https://github.com/boonebgorges/buddypress-group-email-subscription/commit/4acc09d0b16279333a39652f176960c96f61a61e>

Or in this BuddyPress proposal to offload sending at-mentions:

<https://buddypress.trac.wordpress.org/attachment/ticket/6930/6930.wp-async-task-example.patch>

I took a quick look at A5hleyRich's WP_Background_Process class and there's a bit more manual work to do for set up. However, he does have a queue system in place, but his queue system uses the database, which can be slow.

#9 - 2016-06-25 02:31 PM - Daniel Jones

Thank you Ray! I like the TechCrunch library.

I've been messing around with it for a little while, but I'm still having trouble. It seems like my create_duplicate_topics() function just isn't running. Here's what I've got:

```
class BPMFP_Async_Topic_Create extends WP_Async_Task {
    protected $action = 'bbp_new_topic_post_extras';

    protected function prepare_data( $data ) {
        $topic_id = $data[0];
        return( array( 'topic_id' => $topic_id ) );
    }

    protected function run_action() {
        $topic_id = $_POST['topic_id'];

        do_action( 'wp_async_' . $this->action, $topic_id );
    }
}
```

And then, in a functions file included on init at priority 10 I've got:

```
function bpmfp_register_async_actions() {
    if( false === class_exists( 'WP_Async_Task' ) ) {
        require( plugin_dir_path( __FILE__ ) . '/wp-async-task.php' );
    }
    if ( false === class_exists( 'BPMFP_Async_Topic_Create' ) ) {
        require( plugin_dir_path( __FILE__ ) . '/class-bpmfp-async-topic-create.php' );
    }
}
```

```
}  
  
    $bpmfp_async_topic_create = new BPMFP_Async_Topic_Create;  
}  
add_action( 'init', 'bpmfp_register_async_actions', 15 );
```

And I changed the action hook for my create_duplicate_topics function from bbp_new_topic_post_extras to wp_async_bbp_new_topic_post_extras.

I've tried to do some debugging, but I'm getting some confusing results. For one thing, when I add some debug messages to the library's launch_on_shutdown() function, which is supposed to run on the 'shutdown' hook, they're getting printed out during the 'launch' function, which is hooked to 'bbp_new_topic_post_extras' and is what adds launch_on_shutdown() to 'shutdown', so I'm not sure what's up there.

Thanks for your help!

#10 - 2016-06-27 04:37 PM - Raymond Hoh

It looks like you're partially on your way there, Dan!

Now that the bpmfp_create_duplicate_topics() function is run asynchronously, you should move any nonce checks to the BPMFP_Async_Topic_Create::prepare_data() class method.

Also since bpmfp_create_duplicate_topics() relies on a few POST variables, you should return these variables in your BPMFP_Async_Topic_Create::prepare_data() class method in the array where you are also returning the topic ID.

POST variables like 'groups-to-post-to', 'bbp_topic_tags', 'bbp_topic_content', and 'bbp_topic_title' should be returned since they are referenced in the bpmfp_create_duplicate_topics() function.

Let me know if that helps get your code running.

#11 - 2016-07-06 02:18 PM - Daniel Jones

Huge help, Ray - thank you!

Here's how I've got it working on my local install:

<https://github.com/cuny-academic-commons/bp-multiple-forum-post/commit/1ad4fe04cf66d4ebd198a513937f07dc4f84ffda>

And here's my attempt to clean the code up some by splitting into a few different files, and add documentation:

<https://github.com/cuny-academic-commons/bp-multiple-forum-post/commit/47f7535f909881be1826315cc0937b6754f5dc1c>

Let me know if this looks okay!

#12 - 2016-07-19 04:15 AM - Raymond Hoh

Sorry for the delay on this, Dan!

I took a look at the BPMFP code and [made some commits](#) to address a few things, namely fatal errors and an issue where the post content was empty in the email when no groups were selected.

Lastly, I needed to add a hook so plugins can do something when BPMFP has deemed that it is okay to send the email. I needed this hook so I could add some code to Reply By Email so BPMFP emails will work with RBE.

I did come across a UX issue, but this isn't really related to this issue. Let's say I'm a member of two groups and I'm subscribed to "All Email" for both groups. Now, I decide to make a multiple forum topic to both of these groups. If members of both groups reply by email to this topic, the discussion will be forked and it will be hard to keep track of which conversation belongs to which group, especially when replying by email. I thought I'd mention this to get some UX feedback.

#13 - 2016-07-19 11:45 AM - Daniel Jones

Those all look great - thank you Ray!

#14 - 2016-08-28 06:38 PM - Daniel Jones

So the issue for this ticket is addressed in the plugin now: <https://github.com/cuny-academic-commons/bp-multiple-forum-post>

But the UX issue that Ray raised does seem like it ought to be addressed. Sam do you have any thoughts?

#15 - 2016-08-31 10:26 AM - Daniel Jones

Sorry, I hadn't realized that Paige is now in the UX role! Paige - what do you think about the issue that Ray raised above?

#16 - 2016-09-07 06:31 PM - Paige Dupont

- File Screen Shot 2016-09-07 at 6.26.27 PM.png added

Daniel Jones wrote:

Sorry, I hadn't realized that Paige is now in the UX role! Paige - what do you think about the issue that Ray raised above?

Sorry for the delayed response Dan! I've been away for the last week and am just catching up with this now!

While I do see Ray's point about the confusion of a forked conversation, I do think our team has set up some fail safes for this issue already. I just tried this situation myself and there is clear distinction of which group forum topic is being replied to at the top of each email (see screen shot) by presenting a link to said group.

The downside is, you get a lot of emails and have to read carefully as to which thread is being addressed but I don't see it being too much of a problem honestly.

I'd be happy to discuss some other notifications options though if we feel that this could become overwhelming though!

Best,
Paige

Files

Screen Shot 2016-09-07 at 6.26.27 PM.png	16.8 KB	2016-09-07	Paige Dupont
--	---------	------------	--------------