Pulp - Story #2371

Use process recycling by default

10/24/2016 10:41 PM - bmbouter

Status:	CLOSED - WONTFIX	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0:00 hour
Sprint/Milestone:	3.0		
Platform Release:		**QA Contact:**	
Blocks Release:		**Complexity:**	
Backwards Incompatible:	Yes	**Smash Test:**	
Groomed:	No	**Verified:**	No
Sprint Candidate:	Yes	**Verification Required:**	No
Tags:		**Sprint:**	

Description

On the 2.y line a feature was introduced which is disabled by default. The feature was process recycling for celery workers. This was done as Issue #2172.

This issue is to update the conf file defaults from 0 to 2 which will enable this feature by default.

Related issues:

Related to Pulp - Story #2172: Memory Improvements with Process Recycling	**CLOSED - CURRENTRELEASE**

History

#1 - 10/24/2016 10:41 PM - bmbouter

This is blocked until the commits from #2172 are merged from master to 3.0-dev branch.

#2 - 10/24/2016 10:42 PM - bmbouter

- Related to Story #2172: Memory Improvements with Process Recycling added

#3 - 10/31/2016 08:53 PM - mhrivnak

Can you comment on why the value of 2 was chosen? Just based on gut reaction, that seems aggressive. In addition to normal overhead of destroying a process and creating a new one, in this case that also means tearing down and re-creating connections to the database and message broker.

As these things go, the price is likely very small on a mostly-idle system, but grows as resource contention occurs. We haven't quantified that total cost, but even facing an unknown (probably small) cost, we do get to choose how often we pay it. Paying the cost almost as often as possible may be a fine choice, but wouldn't be my personal starting point. What's the thinking?

#4 - 11/02/2016 12:48 PM - bmbouter

Almost all tasks in Pulp require a reservation. Each "reservation task" is actually 2 celery tasks to be processed by a worker. The first is the task itself, the second is a task to release the reservation for that task in the database. Because of this a value of 1 would be unproductive.

Why 2? Anecdotally, Pulp tasks processing real data have service times probably have an average on the order of minutes. Even in our dev environments with no-op tasks it takes multiple seconds. The additional delay caused by process recycling is small probably < 0.5 seconds. Even with a conservative average runtime of 60 seconds, a value of 2 would make that runtime 60.5 seconds which is an overhead of 0.8 %.

I also think the common case of Pulp installations is a mostly idle worker so this optimizes on that by aggressively freeing memory since it may not get more work soon.

Note that the parent process is not torn down, so it won't have to establish a new broker connection in most cases. The parent process does most of the broker communication. That overhead of having to make a new db connection is true. Also note that the process recycling is done by re-forking, which causes the subsequent process to not start from scratch in terms of its Python state. For example the Pulp tasking code was already imported by the parent process.

#5 - 05/14/2018 11:43 PM - dalley

- Status changed from NEW to CLOSED - WONTFIX

RQ re-forks for each task, therefore, this issue can be closed.

#6 - 04/25/2019 06:47 PM - daviddavis

- Sprint/Milestone set to 3.0

#7 - 04/26/2019 10:39 PM - bmbouter

- Tags deleted (Pulp 3)