foreman-tasks - Bug #10962

Concurrence of dynflow is bottlenecked at a single process

06/30/2015 03:03 PM - Ivan Necas

Status: Closed
Priority: Normal
Assigned To: Ivan Necas
Category:
Target version:
Difficulty:
Found in release:

<table>
<thead>
<tr>
<th>Start date: 06/30/2015</th>
<th>Due date:</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Done: 0%</td>
<td>Estimated time: 0.00 hour</td>
</tr>
<tr>
<td>Bugzilla link: 1201428</td>
<td></td>
</tr>
<tr>
<td>Pull request: <a href="https://github.com/Katello/katello/pull/5330">https://github.com/Katello/katello/pull/5330</a></td>
<td></td>
</tr>
</tbody>
</table>

Description

Cloned from https://bugzilla.redhat.com/show_bug.cgi?id=1201428

Description of problem:
Dynflow is a single process within Satellite 6/6.1 and can bottleneck other concurrent tasks.

Version-Release number of selected component (if applicable):
Satellite 6.1 Beta

How reproducible:
100%

Steps to Reproduce:
1. Sync 2 equal sized repos concurrently and time
2. Sync 4 equal sized repos concurrently and time
3. Observe that hardware is not a bottleneck and that dynflow is consuming more and more cpu time at the end of the concurrent tasks.
4. Observe that dynflow executor is limited to 1 process and a single cpu resource during concurrent tasks
5. With no hardware bottlenecks, observe increase in sync timing

Actual results:
Dynflow can be a source of additional latency on concurrent tasks.

Expected results:
dynflow executor to scale as more concurrent tasks are conducted such that it does not bottleneck any other tasks. It is also important that dynflow is not allowed to scale to the point where it uses all the system resources as well.

Additional info:
Attached graphs shows cpu usage of dynflow Executor during syncs.
  top graph = dynflow_excutor cpu usage during 8 repo syncs
  middle graph = dynflow_excutor cpu usage during 6 repo syncs
  bottom graph = dynflow_excutor cpu usage during 2 repo syncs

The average timing values for the syncs are:
  2 repos - 142.15
  4 repos - 204.83
  6 repos - 259.62
  8 repos - 317.05

The growth in overall timing was measured from hammer cli command to sync each repository. The above values come from RHEL 6.6

Associated revisions

Revision 13504c09 - 07/03/2015 11:21 AM - Ivan Necas
  Refs #10962 - Convert foreman-tasks to use new dynflow version

Revision adecbf2a - 07/03/2015 02:18 PM - Ivan Necas
Merge pull request #116 from iNecas/new-dynflow

Refs #10962 - Convert foreman-tasks to use new dynflow version

Revision 5286d035 - 07/03/2015 03:55 PM - Ivan Necas
Refs #10962 - enhance connection cleaning

The ActiveRecord active connections cleaning was sometimes called unnecessary (in case the Dynflow persistence call was part of some bigger Rails action), in which case the preliminary cleaning caused transactions to be committed.

We discovered that when running Katello tests, that relied on the transactions to rollback.

Revision 08feb46d - 07/03/2015 04:37 PM - Ivan Necas

Merge pull request #123 from iNecas/enhance-connection-cleaning

Refs #10962 - enhance connection cleaning

History

#1 - 06/30/2015 03:08 PM - The Foreman Bot
- Status changed from New to Ready For Testing
- Pull request https://github.com/Katello/katello/pull/5330 added

#2 - 08/31/2015 03:15 AM - Ohad Levy
should this be closed?

#3 - 08/31/2015 03:37 AM - Ivan Necas
- Status changed from Ready For Testing to Closed