Ruby master - Bug #18453
YJIT breaks Rails collection caching
12/30/2021 05:54 PM - georgeclaghorn (George Claghorn)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>jhawthorn (John Hawthorn)</td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
<tr>
<td>Backport:</td>
<td>2.6: DONTNEED, 2.7: DONTNEED, 3.0: DONTNEED, 3.1: DONE</td>
</tr>
</tbody>
</table>

ruby -v:
ruby 3.1.0p0 (2021-12-25 revision fb4df44d16) +YJIT [x86_64-linux]

Description
A minimal app that demonstrates this issue is available on GitHub. It has:

- One model, Post, with a string title attribute.
- One controller action, posts#index, which fetches Posts in reverse order of creation into @posts.
- A root route to posts#index.
- One view, app/views/posts/index.html.erb, which uses Rails collection rendering and caching as follows:

```html
<ul>
  <li>
    <%= render partial: "posts/post", collection: @posts, cached: true %>
  </li>
</ul>
```

- A partial, app/views/posts/_post.html.erb as follows:

```erb
<% cache post do %>
  <li><%= post.title %></li>
<% end %>
```

I deployed this app to Heroku here. I configured it to use a Redis cache store.

I added 100 posts like so:

```bash
$ heroku run rails c
> 100.times do |i|
  * Post.create! title: "Post #{i + 1}"
* end
```

All requests to the app index, /, show the posts in reverse chronological order as expected, with cold and warm cache:

```
* Post #100
* Post #99
* Post #98
* Post #97
* Post #96
* Post #95
...
```

I enabled YJIT by setting the RUBYOPT environment variable to --yjit --yjit-exec-mem-size=32 and restarted the app server. I cleared the Redis cache with heroku run rails r 'Rails.cache.redis.flushall.

The first request to / with cold cache begins repeating Post #1 after #92:

```
* Post #100
* Post #99
* Post #98
* Post #97
* Post #96
* Post #95
* Post #94
* Post #93
* Post #92
* Post #1
```
All subsequent requests repeat Post #1 100 times:

* Post #1
* Post #1
* Post #1
* Post #1
* Post #1
* Post #1
* Post #1
* Post #1
* Post #1
* Post #1
...

On restart, Post #1 is repeated after #92 for the first request (as in the second-to-last example). Post #1 is repeated 100 times for subsequent requests (as in the last example).

Disabling YJIT and flushing the Redis cache restores the correct behavior.

**Rails version:** GitHub rails/rails, 7-0-stable branch, revision 499f126c03a4114eb649310e65200fe5d894db0

### Associated revisions

**Revision 5414de4b - 01/01/2022 01:30 AM - jhawthorn (John Hawthorn)**

YJIT: Fix SP index with optarg and unordered kwarg

Previously when we were calling a method with an optional argument and multiple keywords arguments which weren't in the order the receiver expected we would use the wrong SP index to rearrange them.

Fixes Bug #18453

**Revision 2640161d - 01/30/2022 10:01 AM - naruse (Yui NARUSE)**

merge revision(s) 5414de4b6e4372af832e338f8eb7a9fe8de17c84: [Backport #18453]

YJIT: Fix SP index with optarg and unordered kwarg

Previously when we were calling a method with an optional argument and multiple keywords arguments which weren't in the order the receiver expected we would use the wrong SP index to rearrange them.

Fixes Bug #18453

---

**History**

#1 - 12/30/2021 06:15 PM - jhawthorn (John Hawthorn)

- Assignee set to jhawthorn (John Hawthorn)

#2 - 12/30/2021 06:51 PM - jhawthorn (John Hawthorn)

Tracked this down to cache_fragment_name (https://github.com/rails/rails/blob/main/actionview/lib/action_view/helpers/cache_helper.rb#L240) getting the wrong args, so probably a bug in kwarg handling.

#3 - 12/30/2021 07:18 PM - jhawthorn (John Hawthorn)

https://github.com/ruby/ruby/pull/5379

#4 - 01/01/2022 01:32 AM - jhawthorn (John Hawthorn)

- Status changed from Open to Closed
YJIT: Fix SP index with optarg and unordered kwarg

Previously when we were calling a method with an optional argument and multiple keywords arguments which weren't in the order the receiver expected we would use the wrong SP index to rearrange them.

Fixes Bug #18453

#5 - 01/01/2022 01:35 AM - jhawthorn (John Hawthorn)
- Backport changed from 2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN to 2.6: DONTNEED, 2.7: DONTNEED, 3.0: DONTNEED, 3.1: REQUIRED

#6 - 01/01/2022 06:58 AM - alanwu (Alan Wu)
Thank you for such a comprehensive bug report!

#7 - 01/30/2022 10:06 AM - naruse (Yui NARUSE)
- Backport changed from 2.6: DONTNEED, 2.7: DONTNEED, 3.0: DONTNEED, 3.1: REQUIRED to 2.6: DONTNEED, 2.7: DONTNEED, 3.0: DONTNEED, 3.1: DONE

ruby_3_1 2640161df5c18d08ec86a0c1b913d4ee99e102a merged revision(s) 5414de4b6e4372af832e338f8eb7a9fe8de17c84.