Ruby - Bug #18929

ruby master looks slower than 3.1 on a micro benchmark of short-lived objects

07/20/2022 08:21 AM - mame (Yusuke Endoh)



Associated revisions

Revision cdbb9b8555b4ddcc4c557f25ad785cae6209478d - 07/21/2022 02:46 PM - peterzhu2118 (Peter Zhu)

[Bug #18929] Fix heap creation thrashing in GC

Before this commit, if we don't have enough slots after sweeping but had pages on the tomb heap, then the GC would frequently allocate and deallocate pages. This is because after sweeping it would set allocatable pages (since there were not enough slots) but free the pages on the tomb heap.

This commit reuses pages on the tomb heap if there's not enough slots after sweeping.

Revision cdbb9b8555b4ddcc4c557f25ad785cae6209478d - 07/21/2022 02:46 PM - peterzhu2118 (Peter Zhu)

[Bug #18929] Fix heap creation thrashing in GC

Before this commit, if we don't have enough slots after sweeping but had pages on the tomb heap, then the GC would frequently allocate and deallocate pages. This is because after sweeping it would set allocatable pages (since there were not enough slots) but free the pages on the tomb heap.

This commit reuses pages on the tomb heap if there's not enough slots after sweeping.

Revision cdbb9b85 - 07/21/2022 02:46 PM - peterzhu2118 (Peter Zhu)

[Bug #18929] Fix heap creation thrashing in GC

Before this commit, if we don't have enough slots after sweeping but had pages on the tomb heap, then the GC would frequently allocate and deallocate pages. This is because after sweeping it would set allocatable pages (since there were not enough slots) but free the pages on the tomb heap.

This commit reuses pages on the tomb heap if there's not enough slots after sweeping.

Revision 0264424d58e0eb3ff6fc42b7b4164b6e3b8ea8ca - 08/15/2022 02:00 PM - peterzhu2118 (Peter Zhu)

Add test for GC thrashing of young object creation

This test will prevent performance regressions like [Bug #18929].

Revision 0264424d58e0eb3ff6fc42b7b4164b6e3b8ea8ca - 08/15/2022 02:00 PM - peterzhu2118 (Peter Zhu)

Add test for GC thrashing of young object creation

This test will prevent performance regressions like [Bug #18929].

Revision 0264424d - 08/15/2022 02:00 PM - peterzhu2118 (Peter Zhu)

Add test for GC thrashing of young object creation

This test will prevent performance regressions like [Bug #18929].

History

#1 - 07/20/2022 05:44 PM - peterzhu2118 (Peter Zhu)

Thanks for bug report and benchmarking the performance over time! I found a bug that causes thrashing in heap page allocation: <u>https://github.com/ruby/ruby/pull/6156</u>.

Before this patch:

ruby 3.2.0dev (2022-07-20T16:32:04Z pz-bug-18929 b25ee69e38) [arm64-darwin21] ruby -ve '50000000.times { Object.new }' 4.34s user 0.13s system 99% cpu 4.517 total

After this patch:

```
ruby 3.2.0dev (2022-07-20T12:40:31Z pz-bug-18929 86d061294d) [arm64-darwin21]
ruby -ve '50000000.times { Object.new }' 3.48s user 0.05s system 99% cpu 3.547 total
```

Ruby 3.1:

ruby 3.1.2p20 (2022-04-12 revision 4491bb740a) [arm64-darwin21] ruby -ve '50000000.times { Object.new }' 3.43s user 0.06s system 99% cpu 3.489 total

#2 - 07/21/2022 02:07 AM - mame (Yusuke Endoh)

Thank you for the quick fix! Your patch seems to make sense to me.

#3 - 07/21/2022 02:46 PM - peterzhu2118 (Peter Zhu)

- Status changed from Open to Closed

Applied in changeset <u>git|cdbb9b8555b4ddcc4c557f25ad785cae6209478d</u>.

[Bug #18929] Fix heap creation thrashing in GC

Before this commit, if we don't have enough slots after sweeping but had pages on the tomb heap, then the GC would frequently allocate and deallocate pages. This is because after sweeping it would set allocatable pages (since there were not enough slots) but free the pages on the tomb heap.

This commit reuses pages on the tomb heap if there's not enough slots after sweeping.

Files

clipboard-202207201713-zs81l.png

18.2 KB

07/20/2022

mame (Yusuke Endoh)