Ruby master - Bug #11174

threads memory leak

05/23/2015 09:16 PM - cvss (Kirill Vechera)

Status:	Open		
Priority:	Normal		
Assignee:	ko1 (Koichi Sasada)		
Target version:			
ruby -v:	2.2.3, 2.2.0, 2.1.0, 2.0.0, 1.9.3	**Backport:**	2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN

Description

There's strong memory growth during intensive thread using.

Script to demonstrate the problem (on x86_64-linux):

```
loop {
    10.times { 1000.times.map { Thread.new { } }.each(&:join) }
    GC.start # not necessary, just to be sure
    puts File.open('/proc/self/status').grep(/VmRSS:/).first
}
```

Running this script shows RSS growing from 45 Mb at the start time to 700 Mb after few minutes.

```
$ ruby thread_memleak4.rb
VmRSS:     45036 kB
VmRSS:     66748 kB
VmRSS:     87024 kB
...
VmRSS:    678052 kB
```

History

#1 - 07/02/2015 03:48 PM - cvss (Kirill Vechera)

- ruby -v changed from 2.2.0 to 2.2.3, 2.2.0, 2.1.0, 2.0.0, 1.9.3

I've checked the bug on older versions and the 2.2-head: leaks all but 1.8.7

```
$ ruby --version
ruby 2.2.3p139 (2015-07-01) [x86_64-linux]
$ ruby /tmp/thread_memleak4.rb
VmRSS:     47152 kB
VmRSS:     72872 kB
VmRSS:     90448 kB
VmRSS:    105752 kB

$ ruby --version
ruby 1.8.7 (2014-01-28 patchlevel 376) [x86_64-linux]
$ ruby /tmp/thread_memleak4.rb
VmRSS:      2496 kB
VmRSS:      3688 kB
VmRSS:      2768 kB
VmRSS:      2536 kB
VmRSS:      2536 kB
```

#2 - 08/29/2017 07:41 AM - parhs (Coding Gorilla)

I am having similar issue. Even calling open3.capture2 which uses threads results into 250mb of rss memory after a day. 2.3.4 and all 2.3.x.. ,2.4.x

#3 - 09/15/2017 02:27 AM - ko1 (Koichi Sasada)

- Assignee set to ko1 (Koichi Sasada)

Sorry I missed it.

#4 - 09/14/2018 09:17 AM - hartator (Julien Khaleghy)

ko1 (Koichi Sasada) wrote:

> Sorry I missed it.

Thanks for the awesome work on MRI.

I still have the same issue on Ruby 2.5.1 (ruby 2.5.1p57 (2018-03-29 revision 63029) [x86_64-linux-gnu]):

```
VmRSS:     60116 kB
VmRSS:     79448 kB
VmRSS:     89768 kB
VmRSS:     99276 kB
VmRSS:    107916 kB
VmRSS:    113728 kB
VmRSS:    118736 kB
VmRSS:    126600 kB
...
after 5 minutes
....
VmRSS:    754552 kB
VmRSS:    757876 kB
VmRSS:    760132 kB
```

Any idea if it's already fixed somewhere else, or you guys still working on it?

#5 - 09/14/2018 11:12 AM - normalperson (Eric Wong)

hartator@gmail.com wrote:

> Issue #11174 has been updated by hartator (Julien Khaleghy).
>
> Any idea if it's already fixed somewhere else, or you guys
> still working on it?

I missed it, too :x (and now I'm too sleepy to continue...).

It seems specific to glibc (tested 2.24-11+deb9 on Debian
stable) and does not happen with jemalloc (3.6.0).

And it is NOT because of memalign fragmentation, either(*); I
disabled memalign usage to fallback to the aligned malloc+free
fallback; but can still reproduce it.

Finally, it needs multiple threads to reproduce the issue,
a single thread is not enough.

(*) https://sourceware.org/bugzilla/show_bug.cgi?id=14581

#6 - 09/14/2018 11:32 AM - normalperson (Eric Wong)

> It seems specific to glibc

That should be "glibc malloc"; and only old versions. The bug
is fixed in glibc 2.28, at least; likely earlier. (glibc takes
forever to build) It's present in 2.24-11+deb9 on Debian.

MALLOC_ARENA_MAX=1 MALLOC_ARENA_TEST=1 does not seem to help.

I forgot one other parameter which IS the culprit in older glibc versions:

MALLOC_MMAP_THRESHOLD_=131072 # or whatever fixed value you want

So the sliding mmap threshold was the problem in glibc; but it's
fixed upstream, at least. Just have to wait for distros to roll
it out...

#7 - 06/17/2019 03:17 PM - cvss (Kirill Vechera)

The problem remains on a manually built glibc 2.28 with ruby 2.7.0dev (2019-06-17T14:25:47Z trunk 801d0d9dd7) [x86_64-linux]. So if the problem is

related to glibc, the 2.28 version has no fix yet.

Files

thread_memleak4.rb	138 Bytes	05/23/2015	cvss (Kirill Vechera)