

Ruby trunk - Bug #11692

[PATCH] Re-enable GC if stack overflow was caught from signal handler

11/15/2015 05:24 PM - gazay (Alex Gaziev)

Status:	Closed	
Priority:	Normal	
Assignee:	ko1 (Koichi Sasada)	
Target version:		
ruby -v:		Backport: 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: DONE

Description

We got ruby application running on our production server and noticed that it regularly crashes with out of memory errors.

After months of investigation, I narrowed the case to the examples (1/2).

After digging ruby sources and running test code, I found out that GC stopped working after recovering from native stack overflow error.

Probably the relevant code appeared in 2.2 <https://github.com/ruby/ruby/commit/0c391a55d3ed4637e17462d9b9b8aa21e64e2340> where `ruby_disable_gc_stress` became `ruby_disable_gc`.

Patches for trunk and 2.2 branches below.

Associated revisions

Revision 95273072 - 11/19/2015 06:54 AM - ko1 (Koichi Sasada)

- `signal.c`: should also clear `ruby_disable_gc`. [Bug #11692]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@52668 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 52668 - 11/19/2015 06:54 AM - ko1 (Koichi Sasada)

- `signal.c`: should also clear `ruby_disable_gc`. [Bug #11692]

Revision 52668 - 11/19/2015 06:54 AM - ko1 (Koichi Sasada)

- `signal.c`: should also clear `ruby_disable_gc`. [Bug #11692]

Revision 52668 - 11/19/2015 06:54 AM - ko1 (Koichi Sasada)

- `signal.c`: should also clear `ruby_disable_gc`. [Bug #11692]

Revision 52668 - 11/19/2015 06:54 AM - ko1 (Koichi Sasada)

- `signal.c`: should also clear `ruby_disable_gc`. [Bug #11692]

Revision 52668 - 11/19/2015 06:54 AM - ko1 (Koichi Sasada)

- `signal.c`: should also clear `ruby_disable_gc`. [Bug #11692]

Revision 387cc516 - 03/28/2016 05:21 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 52668: [Backport #11692]

```
* signal.c: should also clear ruby_disable_gc.  
[Bug #11692]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_2@54340 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 54340 - 03/28/2016 05:21 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 52668: [Backport #11692]

```
* signal.c: should also clear ruby_disable_gc.
```

History

#1 - 11/16/2015 01:46 PM - gazay (Alex Gaziev)

This issue is related: <https://bugs.ruby-lang.org/issues/11593>

#2 - 11/17/2015 07:52 PM - ebeigarts (Edgars Beigarts)

I'm having similar issue when running tests with capybara that starts additional server in a new thread.

If I have some problems in my rails app that raises SystemStackError in the server thread then I am left without a GC and memory just keeps growing and growing, I have tried manually calling GC.start after that, but it doesn't help, GC.stat displays the same number for major/minor.

I was trying to create a simple example, but I run into other issues too.

Here is the example: <https://gist.github.com/ebeigarts/933f1601332609ed33a8>

Do I need to open a new issue for this?

#3 - 11/17/2015 07:53 PM - ebeigarts (Edgars Beigarts)

- File `gc_threads_issue.rb` added

#4 - 11/18/2015 01:52 AM - gazay (Alex Gaziev)

Edgars Beigarts wrote:

Do I need to open a new issue for this?

I think it is the same problem. I wrote small explanation what is happening: <https://gist.github.com/gazay/54da61919b85eb2e0d42>

#5 - 11/19/2015 06:54 AM - ko1 (Koichi Sasada)

- Status changed from Open to Closed

Applied in changeset r52668.

- signal.c: should also clear `ruby_disable_gc`. [Bug #11692]

#6 - 11/19/2015 06:55 AM - ko1 (Koichi Sasada)

- Assignee set to ko1 (Koichi Sasada)

Thank you for your great survey!

#7 - 11/19/2015 08:55 AM - ebeigarts (Edgars Beigarts)

Tried with ruby head, this solves the GC problems now, but it doesn't solve the other problem that is visible if you run my example -

<https://gist.github.com/ebeigarts/44648eb7b2773e102335>

In ruby 2.2 it looks like if a stack overflow is raised in a thread, the thread just dies. I was running with 2.2.0 and not 2.2.3.

In ruby trunk (and also 2.2.3) it looks like if a stack overflow is raised in a thread 2 times then on the 2nd time the whole process just hangs and the only way to stop is kill -9.

#8 - 11/19/2015 01:47 PM - ko1 (Koichi Sasada)

Unfortunately, it is known problem (2nd time machine stack overflow we can not capture correctly).

- 1st machine stack overflow
 - SEGV
 - check machine stack overflow
 - raise an error from signal handler (*1) by `longjmp`.
- 2nd machine stack overflow
 - SEGV
 - signal status is signaling. So OS can not deliver signal correctly...

The correct way is restoring signal status using `sigsetjmp/siglongjmp` at (*1).

However, on Linux 2.x, `siglongjmp` is too slow than `longjmp`, so that we continue to use `longjmp`, at least the last time we had discussed this issue. We can't slow down Ruby interpreter for such a corner case.

However, Linux 2.x is older OS. So that we can change.
(BTW, I'm using Linux 2.6 on several machines)

#9 - 11/20/2015 06:55 AM - gazay (Alex Gaziev)

Koichi, do you plan to do backport to 2.2 version of this patch?

#10 - 12/18/2015 10:09 AM - gazay (Alex Gaziev)

- *File re-enable-GC-if-stack-overflow-was-caught-from.patch added*

As I see, problem wasn't fixed in Ruby 2.2.4. Patch for it in attachment

#11 - 12/18/2015 10:15 AM - vo.x (Vit Ondruch)

- *Backport changed from 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN to 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: REQUIRED*

Backport should be probably requested

#12 - 12/18/2015 10:16 AM - gazay (Alex Gaziev)

Thank you, didn't know that

#13 - 03/28/2016 05:21 PM - nagachika (Tomoyuki Chikanaga)

- *Backport changed from 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: REQUIRED to 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: DONE*

Backported into ruby_2_2 branch at r54340.

Files

example1.rb	311 Bytes	11/15/2015	gazay (Alex Gaziev)
example2.rb	333 Bytes	11/15/2015	gazay (Alex Gaziev)
re-enable-gc-after-stackoverflow-trunk.patch	985 Bytes	11/15/2015	gazay (Alex Gaziev)
re-enable-gc-after-stackoverflow-ruby-2-2.patch	985 Bytes	11/15/2015	gazay (Alex Gaziev)
gc_threads_issue.rb	1.53 KB	11/17/2015	ebeigarts (Edgars Beigarts)
re-enable-GC-if-stack-overflow-was-caught-from.patch	1.5 KB	12/18/2015	gazay (Alex Gaziev)