

## Ruby master - Bug #4289

### Timeouts in threads cause SEGV

01/18/2011 07:23 PM - kosaki (Motohiro KOSAKI)

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Assignee:</b> akr (Akira Tanaka)	
<b>Target version:</b> 1.9.3	
<b>ruby -v:</b> ruby 1.9.3dev (2011-01-18 trunk 30591) [x86_64-linux]	<b>Backport:</b>
<b>Description</b> =begin Derived from [Bug#4266]  Running deadlock_test.rb in [Bug#4266] on trunk makes segfault. git bisect indicate first bad commit is below.  commit d295957957c828588a8ca3c7b8619c7a93be6b5c Author: akr <a href="mailto:akr@b2dd03c8-39d4-4d8f-98ff-823fe69b080e">akr@b2dd03c8-39d4-4d8f-98ff-823fe69b080e</a> Date: Tue Nov 2 22:37:08 2010 +0000  * vm_method.c (rb_clear_cache_by_class): just return if the class has no method. reported by Eric Wong. [ruby-core:32689]  git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@29673 b2dd03c8-39d4-4d8f-98ff-823fe69b080e  Plus, I've confirmed latest trunk + revert d2959579 doesn't makes segfault. =end	
<b>Related issues:</b>	
Related to Ruby master - Bug #4266: Timeouts in threads cause "ThreadError: d...	<b>Closed</b> <b>01/12/2011</b>
Related to Ruby master - Feature #3905: rb_clear_cache_by_class() called ofte...	<b>Closed</b> <b>10/05/2010</b>

#### Associated revisions

##### Revision e4ba4b79 - 04/29/2011 01:29 AM - kosaki (Motohiro KOSAKI)

- vm\_method.c (rb\_clear\_cache\_by\_class): Revert r29673. It made a segmentation fault regression. [Bug #4289][ruby-core:34554].

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@31378 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 31378 - 04/29/2011 01:29 AM - kosaki (Motohiro KOSAKI)

- vm\_method.c (rb\_clear\_cache\_by\_class): Revert r29673. It made a segmentation fault regression. [Bug #4289][ruby-core:34554].

##### Revision 31378 - 04/29/2011 01:29 AM - kosaki (Motohiro KOSAKI)

- vm\_method.c (rb\_clear\_cache\_by\_class): Revert r29673. It made a segmentation fault regression. [Bug #4289][ruby-core:34554].

##### Revision 31378 - 04/29/2011 01:29 AM - kosaki (Motohiro KOSAKI)

- vm\_method.c (rb\_clear\_cache\_by\_class): Revert r29673. It made a segmentation fault regression. [Bug #4289][ruby-core:34554].

##### Revision 31378 - 04/29/2011 01:29 AM - kosaki (Motohiro KOSAKI)

- vm\_method.c (rb\_clear\_cache\_by\_class): Revert r29673. It made a segmentation fault regression. [Bug #4289][ruby-core:34554].

## Revision 31378 - 04/29/2011 01:29 AM - kosaki (Motohiro KOSAKI)

- `vm_method.c (rb_clear_cache_by_class)`: Revert r29673. It made a segmentation fault regression. [Bug #4289][ruby-core:34554].

## Revision 31378 - 04/29/2011 01:29 AM - kosaki (Motohiro KOSAKI)

- `vm_method.c (rb_clear_cache_by_class)`: Revert r29673. It made a segmentation fault regression. [Bug #4289][ruby-core:34554].

## History

---

### #1 - 04/06/2011 07:36 AM - normalperson (Eric Wong)

- File `0001-timeout.rb-avoid-introducing-new-class-for-every-tim.patch` added

=begin

Hiding, but not fixing the issue is the attached patch:

[PATCH] `timeout.rb`: avoid introducing new class for every timeout

This is expensive because of clearing the method cache upon GC.

As a side effect, it also seems to pass the `deadlock_test.rb[1]` for Bug #4266[2] and also the JRuby `load_timeout.rb[3]` test. However, DO NOT consider this a fix for Bug #4266 or other timeout-related issues. I believe this patch merely hides the real bug and makes it hard to trigger from Ruby standard library.

[1] [http://redmine.ruby-lang.org/attachments/download/1404/deadlock\\_test.rb](http://redmine.ruby-lang.org/attachments/download/1404/deadlock_test.rb)

[2] <http://redmine.ruby-lang.org/issues/4266>

[3] [https://github.com/jruby/jruby/raw/master/test/load/load\\_timeout.rb](https://github.com/jruby/jruby/raw/master/test/load/load_timeout.rb)

=end

### #2 - 04/06/2011 09:18 AM - normalperson (Eric Wong)

=begin

also see

((<[ruby-core:35622]|URL:<http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-core/35622>>))

(redmine doesn't seem to handle mail replies correctly)

=end

### #3 - 04/06/2011 09:23 AM - normalperson (Eric Wong)

=begin

Motohiro KOSAKI wrote:

Running `deadlock_test.rb` in [Bug#4266] on trunk makes segfault. git bisect indicate first bad commit is below.

---

commit d295957957c828588a8ca3c7b8619c7a93be6b5c

Author: akr [akr@b2dd03c8-39d4-4d8f-98ff-823fe69b080e](mailto:akr@b2dd03c8-39d4-4d8f-98ff-823fe69b080e)

Date: Tue Nov 2 22:37:08 2010 +0000

```
* vm_method.c (rb_clear_cache_by_class): just return if the class has
no method. reported by Eric Wong. [ruby-core:32689]
```

```
git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@29673 b2dd03c8-39d4-4d8f-98ff-823fe69b080e
```

---

Plus, I've confirmed latest trunk + revert d2959579 doesn't makes segfault.

Yes, r29673 is bad, I think. The method cache caches for the subclass even if the method belongs to a superclass. I confirmed it with the following debug patch that writes to `stderr` whenever a method-less class is cleared:

```
diff --git a/vm_method.c b/vm_method.c
```

```
index 278941a..021b703 100644
```

```
--- a/vm_method.c
```

```
+++ b/vm_method.c
@@ -84,9 +84,10 @@ void
rb_clear_cache_by_class(VALUE klass)
{
  struct cache_entry *ent, *end;
```

- int nr\_cleared = 0, check\_empty = 0;
- if (RCLASS\_M\_TBL(klass)->num\_entries == 0)
- ██████████
- check\_empty = 1;
- rb\_vm\_change\_state();

```
@@ -98,9 +99,12 @@ rb_clear_cache_by_class(VALUE klass)
if (ent->klass == klass || (ent->me && ent->me->klass == klass)) {
  ent->me = 0;
  ent->mid = 0;
```

- ++nr\_cleared; } ent++; }
- if (check\_empty && nr\_cleared)
- fprintf(stderr, "cleared %d methods for method-less class\n", nr\_cleared); }

VALUE

--

Eric Wong

=end

**#4 - 04/07/2011 08:45 AM - normalperson (Eric Wong)**

- File 0001-revert-r29673-optimization-which-caused-segfaults.patch added

- File 0002-error.c-rb\_mod\_sys\_fail-use-subclass-and-cache.patch added

=begin

Attached are patches to revert r29673 and instate the rb\_mod\_sys\_fail class cache I proposed in [ruby-core:32508]

=end

**#5 - 04/07/2011 10:55 AM - normalperson (Eric Wong)**

=begin

I noticed 0002-error.c-rb\_mod\_sys\_fail-use-subclass-and-cache.patch breaks on latest trunk, actually.

=end

**#6 - 04/07/2011 11:44 AM - normalperson (Eric Wong)**

- File 0001-test-socket-test\_unix-fix-test-failures-from-rb\_mod\_.patch added

=begin

0002-error.c-rb\_mod\_sys\_fail-use-subclass-and-cache.patch breaks existing test cases that use assert\_raise/assert\_raises. This may be a dealbreaker for the patch, unfortunately...

Actual code that uses "rescue Errno::EAGAIN" is unaffected. Anyways I have a patch to fix the failing test case I encountered if breaking existing Test::Unit code is alright...

=end

**#7 - 04/09/2011 03:54 PM - normalperson (Eric Wong)**

- File 0002-introduce-ephemeral-class-flag-for-short-lived-class.patch added

- File 0003-vm\_method.c-ephemeral-classes-do-not-write-expire-ca.patch added

- File 0003-vm\_method.c-ephemeral-classes-do-not-write-expire-ca.patch added

=begin

I found a way to fix the issue without breaking user-facing code and still keep performance :D

I just use a flag to mark a singleton class as ephemeral and have the method cache bypass caching (and expiry) of short-lived ephemeral classes.

The series should be:

- 0001-revert-r29673-optimization-which-caused-segfaults.patch
- 0002-introduce-ephemeral-class-flag-for-short-lived-class.patch
- 0003-vm\_method.c-ephemeral-classes-do-not-write-expire-ca.patch
- 0004-IO-Wait-able-extended-singleton-classes-are-ephemera.patch

Please ignore the following as noise:

- 0002-error.c-rb\_mod\_sys\_fail-use-subclass-and-cache.patch
- 0001-test-socket-test\_unix-fix-test-failures-from-rb\_mod\_.patch

If you use git, I am tracking this my "method-cache-clear" branch in my repo: `git pull git://bogomips.org/ruby method-cache-clear`

I still think 0001-timeout.rb-avoid-introducing-new-class-for-every-tim.patch will be useful, but less important. The Timeout::ExitException subclass cannot be marked as ephemeral from Ruby code...

=end

#### #8 - 04/19/2011 09:23 AM - normalperson (Eric Wong)

=begin

Eric Wong [redmine@ruby-lang.org](mailto:redmine@ruby-lang.org) wrote:

The series should be:

- 0001-revert-r29673-optimization-which-caused-segfaults.patch

Can we get this reversion ASAP since it's confirmed to be causing segfaults in trunk?

Take your time with reviewing the rest, they're low-priority performance improvements (I'll split out to a new ticket if needed):

- 0002-introduce-ephemeral-class-flag-for-short-lived-class.patch
- 0003-vm\_method.c-ephemeral-classes-do-not-write-expire-ca.patch
- 0004-IO-Wait-able-extended-singleton-classes-are-ephemera.patch

Ignore the following sentence:

I still think 0001-timeout.rb-avoid-introducing-new-class-for-every-tim.patch will be useful, but less important.

I realized creating a new class every time is needed to do nested timeouts :->

--

Eric Wong

=end

#### #9 - 04/29/2011 10:32 AM - kosaki (Motohiro KOSAKI)

=begin

0001-revert-r29673-optimization-which-caused-segfaults.patch was committed as r31378.

I think other patch need to get Tanaka-san's review.

=end

#### #10 - 06/14/2011 08:41 AM - akr (Akira Tanaka)

- Status changed from Assigned to Closed

### Files

0001-timeout.rb-avoid-introducing-new-class-for-every-tim.patch	1.33 KB	04/06/2011	normalperson (Eric Wong)
0001-revert-r29673-optimization-which-caused-segfaults.patch	774 Bytes	04/07/2011	normalperson (Eric Wong)

0002-error.c-rb_mod_sys_fail-use-subclass-and-cache.patch	2.08 KB	04/07/2011	normalperson (Eric Wong)
0001-test-socket-test_unix-fix-test-failures-from-rb_mod_.patch	1.25 KB	04/07/2011	normalperson (Eric Wong)
0002-introduce-ephemeral-class-flag-for-short-lived-class.patch	737 Bytes	04/09/2011	normalperson (Eric Wong)
0003-vm_method.c-ephemeral-classes-do-not-write-expire-ca.patch	980 Bytes	04/09/2011	normalperson (Eric Wong)
0003-vm_method.c-ephemeral-classes-do-not-write-expire-ca.patch	980 Bytes	04/09/2011	normalperson (Eric Wong)