

Ruby - Bug #13772

Memory leak recycling stacks for threads in 2.4.1

07/26/2017 08:55 PM - sam.saffron (Sam Saffron)

Status:	Closed	
Priority:	Normal	
Assignee:	ko1 (Koichi Sasada)	
Target version:		
ruby -v:	2.4.1	Backport: 2.2: DONTNEED, 2.3: DONTNEED, 2.4: DONE
Description Per: https://github.com/rest-client/rest-client/issues/611 gem install rest-client <pre>100000.times.each_slice(32) do slice slice.map { Thread.new { RestClient.get('echo.jsontest.com/key/value/one/two') } }.each(&:join) } GC.start(full_mark: true, immediate_sweep: true) slots = GC.stat[:heap_live_slots] puts "slots #{slots} #{i+=1} threads #{Thread.list.count}"</pre> end Causes unbound memory growth on 2.4.1, this was not the case in earlier versions of Ruby or Ruby master. When running through heaptrack and massif-visualizer it looks like all unreclaimed allocations come from thread_recycle_stack Ruby heaps do not grow during tests, all the growth is around unmanaged memory. I feel getting this sorted is urgent, we have seen similar leaks at Discourse.		
Related issues: Related to Ruby - Feature #12628: change block/env structs Closed Related to Ruby - Bug #13775: Ruby hangs when calling scope and belongs_to ma... Closed		

Associated revisions

Revision 92f7813ae27ca7037f9ab6e5744675c01b3fa574 - 08/02/2017 12:50 AM - ko1 (Koichi Sasada)

release VM stack properly.

- cont.c: r55766 change the handling method of Fiber's VM stack. Resumed Fiber points NULL as VM stack and running Thread has responsibility to manage it (marking and releasing).

However, thread_start_func_2()@thread.c and thread_free()@vm.c doesn't free the VM stack if corresponding root Fiber is exist. This causes memory leak. [Bug #13772]
- cont.c (root_fiber_alloc): fib->cont.saved_thread.ec.stack should be NULL because running thread has responsibility to manage this stack.
- vm.c (rb_thread_recycle_stack_release): assert given stack is not NULL (callers should care it).

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@59462 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 92f7813a - 08/02/2017 12:50 AM - ko1 (Koichi Sasada)

release VM stack properly.

- cont.c: r55766 change the handling method of Fiber's VM stack.

Resumed Fiber points NULL as VM stack and running Thread has responsibility to manage it (marking and releasing).

However, `thread_start_func_2()@thread.c` and `thread_free()@vm.c` doesn't free the VM stack if corresponding root Fiber is exist. This causes memory leak. [Bug #13772]

- `cont.c (root_fiber_alloc)`: `fib->cont.saved_thread.ec.stack` should be NULL because running thread has responsibility to manage this stack.
- `vm.c (rb_thread_recycle_stack_release)`: assert given stack is not NULL (callers should care it).

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@59462 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 01cfae3beb0b649ca2ae879825349cf0330a3549 - 08/05/2017 06:35 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 59462,59474: [Backport #13772]

```
release VM stack properly.
```

```
* cont.c: r55766 change the handling method of Fiber's VM stack.
Resumed Fiber points NULL as VM stack and running Thread has
responsibility to manage it (marking and releasing).
```

```
However, thread_start_func_2()@thread.c and thread_free()@vm.c
doesn't free the VM stack if corresponding root Fiber is exist.
This causes memory leak. [Bug #13772]
```

```
* cont.c (root_fiber_alloc): fib->cont.saved_thread.ec.stack should be NULL
because running thread has responsibility to manage this stack.
```

```
* vm.c (rb_thread_recycle_stack_release): assert given stack is not NULL
(callers should care it).
```

```
fix stack storing for root fibers.
```

```
* cont.c (root_fiber_alloc): this function is called by fiber_current()
and fiber_store(). fiber_current() should clear VM stack information
in a fiber data because running thread knows stack information and has
responsibility to manage it. However fiber_store() requires to remain
VM stack information in a fiber data because the responsibility to manage
VM stack is moved to the Fiber from the Thread (and switch to another
fiber).
```

```
* cont.c (root_fiber_alloc): save thread's fiber and root_fiber information.
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_4@59516 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 01cfae3b - 08/05/2017 06:35 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 59462,59474: [Backport #13772]

```
release VM stack properly.
```

```
* cont.c: r55766 change the handling method of Fiber's VM stack.
Resumed Fiber points NULL as VM stack and running Thread has
responsibility to manage it (marking and releasing).
```

```
However, thread_start_func_2()@thread.c and thread_free()@vm.c
doesn't free the VM stack if corresponding root Fiber is exist.
This causes memory leak. [Bug #13772]
```

```
* cont.c (root_fiber_alloc): fib->cont.saved_thread.ec.stack should be NULL
because running thread has responsibility to manage this stack.
```

```
* vm.c (rb_thread_recycle_stack_release): assert given stack is not NULL
(callers should care it).
```

```
fix stack storing for root fibers.
```

```
* cont.c (root_fiber_alloc): this function is called by fiber_current()
and fiber_store(). fiber_current() should clear VM stack information
in a fiber data because running thread knows stack information and has
```

```
responsibility to manage it. However fiber_store() requires to remain
VM stack information in a fiber data because the responsibility to manage
VM stack is moved to the Fiber from the Thread (and switch to another
fiber).
```

```
* cont.c (root_fiber_alloc): save thread's fiber and root_fiber information.
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_4@59516 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 07/27/2017 04:17 PM - sam.saffron (Sam Saffron)

OK I have a standalone repro here: <https://gist.github.com/SamSaffron/f43996d5989cefbfbf69e9557ef13b23>

Also impacts Ruby head

```
i = 0

def parse_param(s)
  unless block_given?
    return enum_for(__method__, s)
  end
  yield "a"
  nil
end

def parse_line(line)
  parts = parse_param('; ' + line)
  parts.next
end

while i < 10_000
  Thread.new { parse_line("Content-Type:application/json; charset=ISO-8859-1") }.join
  slots = GC.stat[:heap_live_slots]
  _, rss = `ps ax -o pid,rss | grep -E "^[[:space:]]*#{$$}"`.strip.split.map(&:to_i)
  puts "slots #{slots} #{i+=1} threads #{Thread.list.count} rss #{rss}"
end

puts "done #{RUBY_VERSION}"
gets

##
# Ruby 2.3.3
# slots 19183, rss 36980

# Ruby 2.4.1
# slots 14448, rss 132572
#
# Ruby latest
# slots 13783 rss 257108
```

#2 - 07/27/2017 07:02 PM - sam.saffron (Sam Saffron)

Looks like ko1 broke this in [#12628](#) :

<https://github.com/ruby/ruby/commit/9f60791a0412cae804f13ed3e746a634c7a0731b>

just bisected it and hit this commit

#3 - 07/31/2017 02:28 PM - sam.saffron (Sam Saffron)

On Friday I downgraded one of our clusters, this is how it looks.

DGEfFuaWAAALSvl.jpg:large

#4 - 07/31/2017 11:59 PM - hsbts (Hiroshi SHIBATA)

- Status changed from Open to Assigned

- Assignee set to ko1 (Koichi Sasada)

#5 - 08/01/2017 01:21 AM - nobu (Nobuyoshi Nakada)

- Related to Feature #12628: change block/env structs added

#6 - 08/01/2017 01:21 AM - nobu (Nobuyoshi Nakada)

- Related to Bug #13775: Ruby hangs when calling scope and belongs_to many times (with mongomapper) added

#7 - 08/01/2017 05:08 AM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN to 2.2: DONTNEED, 2.3: DONTNEED, 2.4: REQUIRED

I'm going to set Backport field based on the report at <https://bugs.ruby-lang.org/issues/13772#note-1>

#8 - 08/02/2017 12:50 AM - ko1 (Koichi Sasada)

- Status changed from Assigned to Closed

Applied in changeset trunk|r59462.

release VM stack properly.

- cont.c: r55766 change the handling method of Fiber's VM stack. Resumed Fiber points NULL as VM stack and running Thread has responsibility to manage it (marking and releasing).

However, thread_start_func_2()@thread.c and thread_free()@vm.c doesn't free the VM stack if corresponding root Fiber is exist. This causes memory leak. [Bug #13772]
- cont.c (root_fiber_alloc): fib->cont.saved_thread.ec.stack should be NULL because running thread has responsibility to manage this stack.
- vm.c (rb_thread_recycle_stack_release): assert given stack is not NULL (callers should care it).

#9 - 08/05/2017 06:35 AM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.2: DONTNEED, 2.3: DONTNEED, 2.4: REQUIRED to 2.2: DONTNEED, 2.3: DONTNEED, 2.4: DONE

ruby_2_4 r59516 merged revision(s) 59462,59474.

#10 - 09/11/2017 05:35 AM - perlun (Per Lundberg)

This bug unfortunately prevents my organization from using 2.4.1 at the moment, we will have to downgrade to 2.3.4. Is there any chance we could get a bug fix release (2.4.2) out including this fix, in the near future? We could always compile from source but it would be great if there would be an "official" fix release including this.

Thanks in advance.

#11 - 09/14/2017 05:31 PM - jraffanie (Joe Rafaniello)

perlun (Per Lundberg) wrote:

This bug unfortunately prevents my organization from using 2.4.1 at the moment, we will have to downgrade to 2.3.4. Is there any chance we could get a bug fix release (2.4.2) out including this fix, in the near future? We could always compile from source but it would be great if there would be an "official" fix release including this.

Thanks in advance.

The fix for this issue was backported [1] and released in 2.4.2 [2]

[1] <https://github.com/ruby/ruby/commit/01cfae3beb0b649ca2ae879825349cf0330a3549>

[2] https://github.com/ruby/ruby/compare/v2_4_1...v2_4_2

#12 - 09/15/2017 09:32 AM - perlun (Per Lundberg)

Thanks a lot, great news!