

Ruby - Bug #14131

test_backtrace.rb: Fails on ppc64le

11/27/2017 04:31 PM - leitao (Breno Leitao)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	Backport:	2.3: UNKNOWN, 2.4: UNKNOWN
Description		
Current test_backtrace.rb fails on ppc64le environment with the following stack:		
<pre>unstable [] ruby git:(trunk) [] ./miniruby -I./lib -I. -I.ext/common ./tool/runruby.rb --extout=.ext -- --disable-gems ./test/runner.rb" --ruby=./miniruby -I./lib -I. -I.ext/common ./tool/runruby.rb --extout=.ext -- --disable-gems" --excludes-dir=./test/excludes --name=test_caller_lev test/ruby/test_backtrace.rb Run options: "--ruby=./miniruby -I./lib -I. -I.ext/common ./tool/runruby.rb --extout=.ext -- --disable-gems" --excludes-dir=./test/excludes --name=test_caller_lev # Running tests: [1/1] TestBacktrace#test_caller_lev = 0.00 s 1) Error: TestBacktrace#test_caller_lev: SystemStackError: stack level too deep /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:93:in `block (2 levels) in test_caller_lev' /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:92:in `times' /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:92:in `block in test_caller_lev' /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:93:in `block (2 levels) in test_caller_lev' /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:92:in `times' /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:92:in `block in test_caller_lev' /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:93:in `block (2 levels) in test_caller_lev' /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:92:in `times' /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:92:in `block in test_caller_lev' /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:93:in `block (2 levels) in test_caller_lev' /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:92:in `times' /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:92:in `block in test_caller_lev' /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:93:in `block (2 levels) in test_caller_lev' /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:92:in `times' /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:92:in `block in test_caller_lev' /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:93:in `block (2 levels) in test_caller_lev' /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:92:in `times' /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:92:in `block in test_caller_lev' /home/breno/ruby/ruby/test/ruby/test_backtrace.rb:106:in `block in test_caller_lev' Finished tests in 0.005963s, 167.7020 tests/s, 1341.6160 assertions/s. 1 tests, 8 assertions, 0 failures, 1 errors, 0 skips ruby -v: ruby 2.5.0dev (2017-11-27 trunk 60922) [powerpc64le-linux]</pre>		
Also, the bisect points to the following commit:		
<pre>commit c4e2cf466448f4283fd3f8a17a73f5fa9b745fe1 Author: normal <normal@b2dd03c8-39d4-4d8f-98ff-823fe69b080e> Date: Thu Jun 8 20:58:03 2017 +0000 tool/runruby.rb: test with smallest possible machine stack Lets ensure none of our C functions use too much stack space and</pre>		

fix all excessive stack usage before releasing the next version.
Reducing C stack usage should reduce conservative GC scanning
time and improve performance.

If there are platform-dependent test failures; excessive stack
usage should be fixed; rather than increasing minimum values or
removing these envs from testing.

Looking at the code, I also found that align the stack to 64kb (size of the page in ppc64le) solves the problem also, something like:

Author: Breno Leitao <breno.leitao@gmail.com>

Date: Mon Nov 27 10:47:00 2017 -0500

stack: align to 64kb

This is a workaround for ppc64le which uses 64kb page size. This align
the stack to the page size. It shouldn't have issues with amd64, but I
am not the expert.

Signed-off-by: Breno Leitao <leitao@debian.org>

```
diff --git a/vm_core.h b/vm_core.h
index 41663fd43e..71f5ee5ba2 100644
--- a/vm_core.h
+++ b/vm_core.h
@@ -576,7 +576,7 @@ typedef struct rb_vm_struct {

/* default values */

-#define RUBY_VM_SIZE_ALIGN 4096
+#define RUBY_VM_SIZE_ALIGN 4096 * 16

#define RUBY_VM_THREAD_VM_STACK_SIZE      ( 128 * 1024 * sizeof(VALUE)) /* 512 KB or 1024 KB
*/
#define RUBY_VM_THREAD_VM_STACK_SIZE_MIN   ( 2 * 1024 * sizeof(VALUE)) /* 8 KB or 16 KB
*/
```

Anyway, any idea on how to solve this issue?

This bug showed up initially in the Debian as <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=881772>

Related issues:

Is duplicate of Ruby - Bug #13757: TestBacktrace#test_caller_lev segaults on PPC

Closed

History

#1 - 11/28/2017 07:51 AM - vo.x (Vit Ondruch)

- Is duplicate of Bug #13757: TestBacktrace#test_caller_lev segaults on PPC added

#2 - 11/28/2017 07:54 AM - vo.x (Vit Ondruch)

- Status changed from Open to Closed

Hi Breno, this is duplicate of [#13757](#). Could you please check the discussion there? I didn't have enough free cycles lately to continue to dig into this ..

#3 - 11/28/2017 04:57 PM - leitao (Breno Leitao)

I was able to narrow down this issue to the following code:

```
max = 20
rec = lambda{|n|

  if n > 0
    rec[n-1]
  end
}
```

```
#rec[max]
Fiber.new{
  rec[max]
}.resume
```

That you should call with:

```
# ./miniruby tool/runruby.rb lambda.rb
```

Some further findings:

- The problem is not reproducible if we use functions recursion instead of lambda recursion.
- The problem does not happen if we run the recursion outside of a Fiber.