Ruby - Bug #4911

timer_thread_function() が thead unsafe

06/21/2011 10:33 AM - kosaki (Motohiro KOSAKI)

Status:	Closed		
Priority:	Normal		
Assignee:	kosaki (Motohiro KOSAKI)		
Target version:	2.0.0		
ruby -v:	trunk	**Backport:**	

Description

どうして、これを先月気づかなかったのだろう。という罪悪感があるのですが、
target 1.9.x で起票します。

timer_thread_function()で vm->running_thread にアクセスするのは
thread unsafe な気がします

０．スレッドＡがrunning_threadである
１．タイマースレッドがvm->running_threadをレジスタにのせる
（レジスタにスレッドＡのアドレスがのっかる）
２．コンテキストスイッチ
３．スレッドＡが終了。スレッドＢがrunning_threadになる
４．スレッドＡの rb_thead_t がfreeされる
５．コンテキストスイッチ
６．そんなｋとはつゆしらず、タイマースレッドはスレッドＡの
アドレスに対して th->interrupt_flag |= 1; するのでメモリ破壊

---

static void
timer_thread_function(void *arg)
{
rb_vm_t *vm = GET_VM(); /* TODO: fix me for Multi-VM */

```
/* for time slice */
RUBY_VM_SET_TIMER_INTERRUPT(vm->running_thread);       ※ここ

/* check signal */
rb_threadptr_check_signal(vm->main_thread);
```

}

Associated revisions

Revision 15b25acd2541ae37f4f874b50128d34d8b079457 - 11/30/2012 01:52 PM - kosaki (Motohiro KOSAKI)

- vm_core.h (rb_vm_struct): add thread_destruct_lock field.
- thread.c (Init_Thread): ditto.
- thread.c (rb_vm_gvl_destroy): ditto.
- thread.c (thread_start_func_2): make sure vm->running_thread don't point to dead thread.
- thread.c (timer_thread_function): close a race against thead destruction. [Bug #4911][ruby-dev:43859]
- vm_core.h (rb_thread_set_current): reset running time of current thread instead of previous thread. We no longer assume previous running thread still live.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@38063 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 15b25acd - 11/30/2012 01:52 PM - kosaki (Motohiro KOSAKI)

- vm_core.h (rb_vm_struct): add thread_destruct_lock field.
- thread.c (Init_Thread): ditto.
- thread.c (rb_vm_gvl_destroy): ditto.
- thread.c (thread_start_func_2): make sure vm->running_thread don't point to dead thread.
- thread.c (timer_thread_function): close a race against thead destruction. [Bug #4911][ruby-dev:43859]
- vm_core.h (rb_thread_set_current): reset running time of current thread instead of previous thread. We no longer assume previous running thread still live.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@38063 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 03/18/2012 06:46 PM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Assigned

#2 - 09/14/2012 05:34 PM - kosaki (Motohiro KOSAKI)

- Assignee changed from kosaki (Motohiro KOSAKI) to ko1 (Koichi Sasada)

I've made a patch.

https://github.com/ruby/ruby/pull/182

ko1, could you please review it?

#3 - 09/22/2012 07:10 AM - ko1 (Koichi Sasada)

問題はわかったのですが，
これがどうして解決するのかわかりません．
（結局，＋ if (!vm->running_thread->yielding) の部分で running_thread を触っているように見える）

解説してもらえないでしょうか．

では，どうやって解決するかというと難しいですね．タイマスレッドが動いている間は free しない，とか，そういうのになりそうな感じ．

#4 - 09/22/2012 07:32 AM - ko1 (Koichi Sasada)

(2012/09/21 15:10), ko1 (Koichi Sasada) wrote:

> では，どうやって解決するかというと難しいですね．タイマスレッドが動いている間は free しない，とか，そういうのになりそうな感じ．

勘で書いてみたんですが，こんな感じでしょうか．再現が出来なさそうなので，
テストの書きようが無さそう．

Index: vm_core.h

--- vm_core.h (revision 37007)
+++ vm_core.h (working copy)
@@ -312,6 +312,7 @@ typedef struct rb_vm_struct {
int thread_abort_on_exception;
unsigned long trace_flag;
volatile int sleeper;

- volatile struct rb_thread_struct *timer_thread_target;

  /* object management */
  VALUE mark_object_ary;
  Index: thread.c

===
--- thread.c (revision 37007)
+++ thread.c (working copy)
@@ -3388,9 +3388,13 @@ timer_thread_function(void *arg)
rb_vm_t *vm = GET_VM(); /* TODO: fix me for Multi-VM */

/* for time slice */

- vm->timer_thread_target = vm->running_thread;

- {
  if (!vm->running_thread->yielding) {
  RUBY_VM_SET_TIMER_INTERRUPT(vm->running_thread);
  }

- }

- vm->timer_thread_target = 0;

  /* check signal */
  rb_threadptr_check_signal(vm->main_thread);
  Index: vm.c
  ===
  --- vm.c (revision 37007)
  +++ vm.c (working copy)
  @@ -1724,6 +1724,9 @@ thread_free(void *ptr)
  free(th->altstack);
  }
  #endif

- [redacted]

- [redacted]

- [redacted]
  `ruby_xfree(ptr);`

  }
  if (ruby_current_thread == th)

--
// SASADA Koichi at atdot dot net

#5 - 09/22/2012 09:41 AM - kosaki (Motohiro KOSAKI)

> 問題はわかったのですが，
> これがどうして解決するのかわかりません．
> （結局，+ if (!vm->running_thread->yielding) の部分で running_thread を触っているように見える）
> 解説してもらえないでしょうか．

あ、しまった。放置しすぎていてコンテキストスイッチだけを救うパッチを書いってしまった。本当に問題なのはスイッチ元のスレッドがfreeされていてメモリ破壊に至るケースなのでおっしゃるとおり考慮不足です

#6 - 09/22/2012 09:44 AM - kosaki (Motohiro KOSAKI)

で、Comment#4のパッチだと残念なことにタイマースレッドとスレッド解放ルートに共通のロックがないため、グローバル変数に代入してもメモリバリア的な意味でthread_free() からは可視にならなさそう

#7 - 09/22/2012 01:23 PM - ko1 (Koichi Sasada)

(2012/09/21 17:44), kosaki (Motohiro KOSAKI) wrote:

> で、Comment#4のパッチだと残念なことにタイマースレッドとスレッド解放ルートに共通のロックがないため、グローバル変数に代入してもメモリバリア的な意味でthread_free() からは可視にならなさそう

うーん，なるほど．
真面目に mutex で囲いますかねぇ．それ以外の冴えたやり方はないかなぁ．
別に，ここでオーバヘッドになることもないだろうから，いっか．

--
// SASADA Koichi at atdot dot net

#8 - 11/26/2012 09:22 AM - ko1 (Koichi Sasada)

- Assignee changed from ko1 (Koichi Sasada) to kosaki (Motohiro KOSAKI)

これは忘れてはいけなかった気がする.
小崎先生お願いします.

#9 - 11/30/2012 10:52 PM - kosaki (Motohiro KOSAKI)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r38063.
Motohiro, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- vm_core.h (rb_vm_struct): add thread_destruct_lock field.
- thread.c (Init_Thread): ditto.
- thread.c (rb_vm_gvl_destroy): ditto.
- thread.c (thread_start_func_2): make sure vm->running_thread don't point to dead thread.
- thread.c (timer_thread_function): close a race against thead destruction. [Bug #4911][ruby-dev:43859]
- vm_core.h (rb_thread_set_current): reset running time of current thread instead of previous thread. We no longer assume previous running thread still live.