

Ruby trunk - Feature #9761

TRY_WITH_GC with ruby_gc_stress

04/20/2014 12:49 AM - wanabe (_ wanabe)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		
Description		
I hope TRY_WITH_GC() may use ruby_gc_stress value.		
Currently, ruby_gc_stress doesn't affect simple xmalloc()/xrealloc()/xcalloc(). It is usefull to make these functions under gc-stress. For example, #9657 is difficult to reproduce on trunk because there is not object creation but memory allocation after BigDecimal_new().		
Related issues:		
Related to Ruby trunk - Bug #9859: An object with 6 instance variables causes...	Closed	05/23/2014

Associated revisions

Revision a633ed28 - 04/21/2014 09:53 PM - nobu (Nobuyoshi Nakada)

gc.c: GC.stress after realloc

- gc.c (objspace_malloc_increase): run GC after realloc not only malloc and calloc by GC.stress. [ruby-core:62103] [Feature #9761]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@45653 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 45653 - 04/21/2014 09:53 PM - nobu (Nobuyoshi Nakada)

gc.c: GC.stress after realloc

- gc.c (objspace_malloc_increase): run GC after realloc not only malloc and calloc by GC.stress. [ruby-core:62103] [Feature #9761]

Revision 45653 - 04/21/2014 09:53 PM - nobu (Nobuyoshi Nakada)

gc.c: GC.stress after realloc

- gc.c (objspace_malloc_increase): run GC after realloc not only malloc and calloc by GC.stress. [ruby-core:62103] [Feature #9761]

Revision 45653 - 04/21/2014 09:53 PM - nobu (Nobuyoshi Nakada)

gc.c: GC.stress after realloc

- gc.c (objspace_malloc_increase): run GC after realloc not only malloc and calloc by GC.stress. [ruby-core:62103] [Feature #9761]

Revision 45653 - 04/21/2014 09:53 PM - nobu (Nobuyoshi Nakada)

gc.c: GC.stress after realloc

- gc.c (objspace_malloc_increase): run GC after realloc not only malloc and calloc by GC.stress. [ruby-core:62103] [Feature #9761]

Revision 45653 - 04/21/2014 09:53 PM - nobu (Nobuyoshi Nakada)

gc.c: GC.stress after realloc

- gc.c (objspace_malloc_increase): run GC after realloc not only malloc and calloc by GC.stress. [ruby-core:62103] [Feature #9761]

Revision 45653 - 04/21/2014 09:53 PM - nobu (Nobuyoshi Nakada)

gc.c: GC.stress after realloc

- gc.c (objspace_malloc_increase): run GC after realloc not only malloc and calloc by GC.stress. [ruby-core:62103] [Feature #9761]

Revision 5890cb9d - 04/21/2014 09:54 PM - nobu (Nobuyoshi Nakada)

gc.c: full mark after malloc/realloc

- gc.c (objspace_malloc_increase): run full mark if 0x04 bit is set in ruby_gc_stress. [ruby-core:62103] [Feature #9761]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@45655 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 45655 - 04/21/2014 09:54 PM - nobu (Nobuyoshi Nakada)

gc.c: full mark after malloc/realloc

- gc.c (objspace_malloc_increase): run full mark if 0x04 bit is set in ruby_gc_stress. [ruby-core:62103] [Feature #9761]

Revision 45655 - 04/21/2014 09:54 PM - nobu (Nobuyoshi Nakada)

gc.c: full mark after malloc/realloc

- gc.c (objspace_malloc_increase): run full mark if 0x04 bit is set in ruby_gc_stress. [ruby-core:62103] [Feature #9761]

Revision 45655 - 04/21/2014 09:54 PM - nobu (Nobuyoshi Nakada)

gc.c: full mark after malloc/realloc

- gc.c (objspace_malloc_increase): run full mark if 0x04 bit is set in ruby_gc_stress. [ruby-core:62103] [Feature #9761]

Revision 45655 - 04/21/2014 09:54 PM - nobu (Nobuyoshi Nakada)

gc.c: full mark after malloc/realloc

- gc.c (objspace_malloc_increase): run full mark if 0x04 bit is set in ruby_gc_stress. [ruby-core:62103] [Feature #9761]

Revision 45655 - 04/21/2014 09:54 PM - nobu (Nobuyoshi Nakada)

gc.c: full mark after malloc/realloc

- gc.c (objspace_malloc_increase): run full mark if 0x04 bit is set in ruby_gc_stress. [ruby-core:62103] [Feature #9761]

Revision 45655 - 04/21/2014 09:54 PM - nobu (Nobuyoshi Nakada)

gc.c: full mark after malloc/realloc

- gc.c (objspace_malloc_increase): run full mark if 0x04 bit is set in ruby_gc_stress. [ruby-core:62103] [Feature #9761]

History

#1 - 04/20/2014 12:52 AM - wanabe (_ wanabe)

- Description updated

#2 - 04/20/2014 12:56 AM - nobu (Nobuyoshi Nakada)

- Description updated

- Category set to core

- Status changed from Open to Feedback

Patches are welcome.

#3 - 04/20/2014 01:45 AM - wanabe (_ wanabe)

- File `try_with_gc-with-gc_stress.patch` added

- Category deleted (core)

- Status changed from Feedback to Open

I have written but I have not been able to avoid both temporary assignment and duplication of `garbage_collect_with_gvl()`.

#4 - 04/20/2014 01:46 AM - wanabe (_ wanabe)

- Category set to core

#5 - 04/21/2014 01:05 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Feedback

Seems `objspace_malloc_increase()` already runs GC after `xmalloc` and `xcalloc` if `ruby_gc_stress` is set. You may want it to run after `xrealloc` too?

Or to run full marking?

#6 - 04/21/2014 03:23 PM - wanabe (_ wanabe)

- File *xrealloc-with-gc_stress.patch* added
- Status changed from *Feedback* to *Open*

Nobuyoshi Nakada wrote:

Seems `objspace_malloc_increase()` already runs GC after `xmalloc` and `xcalloc` if `ruby_gc_stress` is set.
You may want it to run after `xrealloc` too?
Or to run full marking?

Sorry for my poor code reading...

It is enough to add a case of `xrealloc` for my concern, reproducing [#9657](#).

Anyone can close/reject this ticket if you think this is far from the original proposal.

#7 - 04/21/2014 09:53 PM - nobu (Nobuyoshi Nakada)

- Status changed from *Open* to *Closed*
- % Done changed from *0* to *100*

Applied in changeset `r45653`.

gc.c: GC.stress after realloc

- `gc.c (objspace_malloc_increase)`: run GC after realloc not only `malloc` and `calloc` by `GC.stress`. [`ruby-core:62103`] [Feature [#9761](#)]

#8 - 05/24/2014 03:52 AM - nobu (Nobuyoshi Nakada)

- Related to Bug [#9859](#): *An object with 6 instance variables causes SEGV* added

Files

<code>try_with_gc-with-gc_stress.patch</code>	691 Bytes	04/20/2014	wanabe (_ wanabe)
<code>xrealloc-with-gc_stress.patch</code>	456 Bytes	04/21/2014	wanabe (_ wanabe)