

Ruby master - Feature #8985

xwillfree - promise to free memory

10/04/2013 08:38 PM - funny_falcon (Yura Sokolov)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	2.1.0
Description	
<p>This patch changes semantic of RUBY_GC_MALLOC_LIMIT. Instead of being "periodical trigger" it becomes more like "safety trigger" which fires in allocation increase (instead of allocation amount). So that there is less need to tune RUBY_GC_MALLOC_LIMIT at all, and default 8Mb becomes meaningful.</p> <p>Before GC relaxation in commit 8c0033a make check ran 13% faster (292s instead of 338s) and doesn't seems to use more memory. It is now runs at the same speed, but I propose to revert some part of GC relaxation.</p> <p>Tradeoffs for patch simplicity:</p> <ul style="list-style-type: none">• it is not exact: only String, Array, Object, Struct, Bignum and Time are handled• only one function (xwillfree) introduced. Perhaps, more readable api could be useful.• xwillfree exposed to the public (ruby.h). Perhaps, it should be in an internal.h, but st.c doesn't include internal.h. And may be it could be useful for extensions. <p>https://github.com/ruby/ruby/pull/414 https://github.com/ruby/ruby/pull/414.patch https://github.com/ruby/ruby/pull/414.diff</p>	

Associated revisions

Revision 8ac4f421 - 10/17/2013 07:57 AM - ko1 (Koichi Sasada)

- gc.c, internal.h: add new internal memory mangement functions.
- void *ruby_xsize_realloc(void *ptr, size_t new_size, size_t old_size)
- void ruby_xsize_free(void *x, size_t size) These functions accept additional size parameter to calculate more accurate malloc_increase parameter which control GC timing. [Feature #8985]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@43330 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 43330 - 10/17/2013 07:57 AM - ko1 (Koichi Sasada)

- gc.c, internal.h: add new internal memory mangement functions.
- void *ruby_xsize_realloc(void *ptr, size_t new_size, size_t old_size)
- void ruby_xsize_free(void *x, size_t size) These functions accept additional size parameter to calculate more accurate malloc_increase parameter which control GC timing. [Feature #8985]

Revision 43330 - 10/17/2013 07:57 AM - ko1 (Koichi Sasada)

- gc.c, internal.h: add new internal memory mangement functions.
- void *ruby_xsize_realloc(void *ptr, size_t new_size, size_t old_size)
- void ruby_xsize_free(void *x, size_t size) These functions accept additional size parameter to calculate more accurate malloc_increase parameter which control GC timing. [Feature #8985]

Revision 43330 - 10/17/2013 07:57 AM - ko1 (Koichi Sasada)

- gc.c, internal.h: add new internal memory mangement functions.
- void *ruby_xsize_realloc(void *ptr, size_t new_size, size_t old_size)
- void ruby_xsize_free(void *x, size_t size) These functions accept additional size parameter to calculate more accurate malloc_increase parameter which control GC timing. [Feature #8985]

Revision 43330 - 10/17/2013 07:57 AM - ko1 (Koichi Sasada)

- gc.c, internal.h: add new internal memory mangement functions.
- void *ruby_xsizerealloc(void *ptr, size_t new_size, size_t old_size)
- void ruby_xsizefree(void *x, size_t size) These functions accept additional size parameter to calculate more accurate malloc_increas parameter which control GC timing. [Feature #8985]

Revision 43330 - 10/17/2013 07:57 AM - ko1 (Koichi Sasada)

- gc.c, internal.h: add new internal memory mangement functions.
- void *ruby_xsizerealloc(void *ptr, size_t new_size, size_t old_size)
- void ruby_xsizefree(void *x, size_t size) These functions accept additional size parameter to calculate more accurate malloc_increas parameter which control GC timing. [Feature #8985]

Revision 43330 - 10/17/2013 07:57 AM - ko1 (Koichi Sasada)

- gc.c, internal.h: add new internal memory mangement functions.
- void *ruby_xsizerealloc(void *ptr, size_t new_size, size_t old_size)
- void ruby_xsizefree(void *x, size_t size) These functions accept additional size parameter to calculate more accurate malloc_increas parameter which control GC timing. [Feature #8985]

History

#1 - 10/04/2013 09:38 PM - funny_falcon (Yura Sokolov)

SASADA Koichi wrote:

Ah, it is synchronicity.

I have another idea to approach for it.

how about another version of ruby_xfree() and ruby_xrealloc() to passing 2nd argument, which is passing same information pasing to xwill_free().

```
ptr = xmalloc2(100); /* allocate 100 byte /
xrealloc2(ptr, 100, 200); / reallocate 100 to 200 /
xfree2(ptr, 200); /* free 200 byte */
```

Doing same objective. but reduce one function call. I like this approach.

I mentioned that one function is tradeoff for patch simplicity - just for idea presentation. And there is REALLOC_N: passing another one argument to could look ugly in several place.

Any way, I like idea with additional argument to functions.

#2 - 10/04/2013 09:53 PM - ko1 (Koichi Sasada)

Ah, it is synchronicity.

I have another idea to approach for it.

how about another version of ruby_xfree() and ruby_xrealloc() to passing 2nd argument, which is passing same information pasing to xwill_free().

```
ptr = xmalloc2(100); /* allocate 100 byte /
xrealloc2(ptr, 100, 200); / reallocate 100 to 200 /
xfree2(ptr, 200); /* free 200 byte */
```

Doing same objective. but reduce one function call. I like this approach.

(2013/10/04 20:38), funny_falcon (Yura Sokolov) wrote:

Issue [#8985](#) has been reported by funny_falcon (Yura Sokolov).

Feature [#8985](#): xwillfree - promise to free memory
<https://bugs.ruby-lang.org/issues/8985>

Author: funny_falcon (Yura Sokolov)
Status: Open
Priority: Normal

Assignee:
Category: core
Target version: current: 2.1.0

This patch changes semantic of RUBY_GC_MALLOC_LIMIT.
Instead of being "periodical trigger" it becomes more like "safety trigger"
which fires in allocation increase (instead of allocation amount).
So that there is less need to tune RUBY_GC_MALLOC_LIMIT at all, and default
8Mb becomes meaningful.

Before GC relaxation in commit 8c0033a make check ran 13% faster
(292s instead of 338s) and doesn't seem to use more memory. It is now
runs at the same speed, but I propose to revert some part of GC
relaxation.

Tradeoffs for patch simplicity:

- it is not exact: only String, Array, Object, Struct, Bignum and Time are handled
- only one function (xwillfree) introduced. Perhaps, more readable api could be useful.
- xwillfree exposed to the public (ruby.h). Perhaps, it should be in an internal.h, but st.c doesn't include internal.h. And maybe it could be useful for extensions.

<https://github.com/ruby/ruby/pull/414>
<https://github.com/ruby/ruby/pull/414.patch>
<https://github.com/ruby/ruby/pull/414.diff>

--
// SASADA Koichi at atdot dot net

#3 - 10/04/2013 09:53 PM - ko1 (Koichi Sasada)

(2013/10/04 21:30), SASADA Koichi wrote:

Ah, it is synchronicity.

because we Heroku ruby team discussing about xfree2 and xrealloc2.

```
ptr = xmalloc2(100); /* allocate 100 byte */
```

Sorry. We don't need xmalloc2().

--
// SASADA Koichi at atdot dot net

#4 - 10/05/2013 10:53 AM - ko1 (Koichi Sasada)

(2013/10/04 21:38), funny_falcon (Yura Sokolov) wrote:

Any way, I like idea with additional argument to functions.

One more advantage of additional argument is we can verify passed size
argument with CALC_EXACT_MALLOC_SIZE option in gc.c.

--
// SASADA Koichi at atdot dot net

#5 - 10/17/2013 04:57 PM - ko1 (Koichi Sasada)

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r43330.
Yura, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

-
- gc.c, internal.h: add new internal memory management functions.
 - void *ruby_xsizedrealloc(void *ptr, size_t new_size, size_t old_size)
 - void ruby_xsizedfree(void *x, size_t size) These functions accept additional size parameter to calculate more accurate malloc_increase

parameter which control GC timing. [Feature [#8985](#)]

Files

xwillfree.diff	11.1 KB	10/04/2013	funny_falcon (Yura Sokolov)
----------------	---------	------------	-----------------------------