

Ruby trunk - Bug #9687

Make a cap for malloc_limit adjustment

03/29/2014 04:14 AM - ko1 (Koichi Sasada)

Status: Closed	
Priority: Normal	
Assignee: ko1 (Koichi Sasada)	
Target version: 2.2.0	
ruby -v: 2.2	Backport: 2.0.0: DONTNEED, 2.1: DONE
Description	
The blog post Ruby Garbage Collection: Still Not Ready for Production pointed out that the following code consumes all of memory:	
<pre>while true "a" * (1024 ** 2) end</pre>	
This is because `malloc_limit` grows and grows infinitely.	
The following patch will solve this issue.	
Index: gc.c	
=====	
--- gc.c (revision 45433)	
+++ gc.c (working copy)	
@@ -2890,7 +2890,7 @@ gc_before_sweep(rb_objspace_t *objspace)	
malloc_limit = (size_t)(inc * gc_params.malloc_limit_growth_factor);	
if (gc_params.malloc_limit_max > 0 && /* ignore max-check if 0 */	
malloc_limit > gc_params.malloc_limit_max) {	
- malloc_limit = inc;	
+ malloc_limit = gc_params.malloc_limit_max;	
}	
}	
else {	
Maybe this patch should be applied for 2.1.x series.	

Associated revisions

Revision 835186a3 - 03/29/2014 04:37 AM - ko1 (Koichi Sasada)

- gc.c (gc_before_sweep): cap malloc_limit to gc_params.malloc_limit_max. It can grow and grow with such case: loop{"a" * (1024 ** 2)} [Bug #9687] This issue is pointed by Tim Robertson.
<http://www.omniref.com/blog/blog/2014/03/27/ruby-garbage-collection-still-not-ready-for-production/>

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@45468 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 45468 - 03/29/2014 04:37 AM - ko1 (Koichi Sasada)

- gc.c (gc_before_sweep): cap malloc_limit to gc_params.malloc_limit_max. It can grow and grow with such case: loop{"a" * (1024 ** 2)} [Bug #9687] This issue is pointed by Tim Robertson.
<http://www.omniref.com/blog/blog/2014/03/27/ruby-garbage-collection-still-not-ready-for-production/>

Revision 45468 - 03/29/2014 04:37 AM - ko1 (Koichi Sasada)

- gc.c (gc_before_sweep): cap malloc_limit to gc_params.malloc_limit_max. It can grow and grow with such case: loop{"a" * (1024 ** 2)} [Bug #9687] This issue is pointed by Tim Robertson.
<http://www.omniref.com/blog/blog/2014/03/27/ruby-garbage-collection-still-not-ready-for-production/>

Revision 45468 - 03/29/2014 04:37 AM - ko1 (Koichi Sasada)

- gc.c (gc_before_sweep): cap malloc_limit to gc_params.malloc_limit_max. It can grow and grow with such case: loop{"a" * (1024 ** 2)} [Bug #9687] This issue is pointed by Tim Robertson.

Revision 45468 - 03/29/2014 04:37 AM - ko1 (Koichi Sasada)

- gc.c (gc_before_sweep): cap malloc_limit' to gc_params.malloc_limit_max. It can grow and grow with such case: loop{"a" * (1024 ** 2)}' [Bug #9687] This issue is pointed by Tim Robertson.
<http://www.omniref.com/blog/blog/2014/03/27/ruby-garbage-collection-still-not-ready-for-production/>

Revision 45468 - 03/29/2014 04:37 AM - ko1 (Koichi Sasada)

- gc.c (gc_before_sweep): cap malloc_limit' to gc_params.malloc_limit_max. It can grow and grow with such case: loop{"a" * (1024 ** 2)}' [Bug #9687] This issue is pointed by Tim Robertson.
<http://www.omniref.com/blog/blog/2014/03/27/ruby-garbage-collection-still-not-ready-for-production/>

Revision 45468 - 03/29/2014 04:37 AM - ko1 (Koichi Sasada)

- gc.c (gc_before_sweep): cap malloc_limit' to gc_params.malloc_limit_max. It can grow and grow with such case: loop{"a" * (1024 ** 2)}' [Bug #9687] This issue is pointed by Tim Robertson.
<http://www.omniref.com/blog/blog/2014/03/27/ruby-garbage-collection-still-not-ready-for-production/>

Revision ac0f8d80 - 05/04/2014 04:17 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) r45468: [Backport #9687]

```
* gc.c (gc_before_sweep): cap `malloc_limit' to
gc_params.malloc_limit_max. It can grow and grow with such case:
`loop{"a" * (1024 ** 2)}'
[Bug #9687]
This issue is pointed by Tim Robertson.
http://www.omniref.com/blog/blog/2014/03/27/ruby-garbage-collection-still-not-ready-for-production/
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_1@45817 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 45817 - 05/04/2014 04:17 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) r45468: [Backport #9687]

```
* gc.c (gc_before_sweep): cap `malloc_limit' to
gc_params.malloc_limit_max. It can grow and grow with such case:
`loop{"a" * (1024 ** 2)}'
[Bug #9687]
This issue is pointed by Tim Robertson.
http://www.omniref.com/blog/blog/2014/03/27/ruby-garbage-collection-still-not-ready-for-production/
```

History

#1 - 03/29/2014 04:37 AM - ko1 (Koichi Sasada)

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

Applied in changeset [r45468](#).

- gc.c (gc_before_sweep): cap malloc_limit' to gc_params.malloc_limit_max. It can grow and grow with such case: loop{"a" * (1024 ** 2)}' [Bug #9687] This issue is pointed by Tim Robertson.
<http://www.omniref.com/blog/blog/2014/03/27/ruby-garbage-collection-still-not-ready-for-production/>

#2 - 03/29/2014 04:43 AM - ko1 (Koichi Sasada)

- Status changed from Closed to Open
- Backport changed from 2.0.0: UNKNOWN, 2.1: UNKNOWN to 2.0.0: UNKNOWN, 2.1: REQUIRED

#3 - 03/30/2014 01:16 PM - nagachika (Tomoyuki Chikanaga)

- Status changed from Open to Closed
- Backport changed from 2.0.0: UNKNOWN, 2.1: REQUIRED to 2.0.0: REQUIRED, 2.1: REQUIRED

#4 - 03/30/2014 02:54 PM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.0.0: REQUIRED, 2.1: REQUIRED to 2.0.0: DONTNEED, 2.1: REQUIRED

#5 - 05/04/2014 04:18 PM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.0.0: DONTNEED, 2.1: REQUIRED to 2.0.0: DONTNEED, 2.1: DONE

[r45468](#) was backported into ruby_2_1 branch at r45817.