

## Ruby - Bug #8983

### [PATCH] GC.stat[:heap\_free\_num] returns number of unused slots on heap

10/03/2013 06:25 PM - tmm1 (Aman Karmani)

<b>Status:</b>	Closed	<b>Backport:</b> 1.9.3: UNKNOWN, 2.0.0: UNKNOWN
<b>Priority:</b>	Normal	
<b>Assignee:</b>	ko1 (Koichi Sasada)	
<b>Target version:</b>	2.1.0	
<b>ruby -v:</b>	ruby 2.1.0dev (2013-10-03 trunk 43129) [x86_64-darwin12.4.1]	
<b>Description</b>		
<p>In ruby 1.9 and 2.0, gc_stat() used to call rest_sweep() so heap_free_num always equals the number of free slots (ObjectSpace.count_objects[:FREE]). In 2.1 the rest_sweep() was removed, so free_num is not always accurate. This patch fixes compatibility with the previous behavior.</p> <p>With this patch, it is also easier to perform "out-of-band" GC in unicorn and passenger. For example: if your application requires 100k objects per request, then you can run the following in-between every request:</p> <pre>GC.start if GC.stat[:heap_free_num] &lt; 100_000</pre> <p>I also added a objspace-&gt;heap.limit because HEAP_OBJ_LIMIT*heap_used is not accurate due to memory re-alignment in heap_assign_slot.</p> <p><a href="https://github.com/tmm1/ruby/compare/gc-stat-free-num">https://github.com/tmm1/ruby/compare/gc-stat-free-num</a> <a href="https://github.com/tmm1/ruby/compare/gc-stat-free-num.patch">https://github.com/tmm1/ruby/compare/gc-stat-free-num.patch</a></p>		

#### Associated revisions

##### Revision a24705cfa52d14470354600832d129c5b8af64a1 - 10/04/2013 10:05 AM - tmm1 (Aman Karmani)

gc.c: add rb\_objspace.limit

```
* gc.c: add rb_objspace.limit to keep accurate count of total heap
slots [ruby-core:57633] [Bug #8983]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@43138 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision a24705cf - 10/04/2013 10:05 AM - tmm1 (Aman Karmani)

gc.c: add rb\_objspace.limit

```
* gc.c: add rb_objspace.limit to keep accurate count of total heap
slots [ruby-core:57633] [Bug #8983]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@43138 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 907a03fee52b06760cec2d6fbd0dd2e693bef970 - 10/04/2013 10:05 AM - tmm1 (Aman Karmani)

gc.c: add objspace\_free\_num and make GC.stat[:heap\_free\_num] use it

```
* gc.c (objspace_free_num): new method for available/free slots on
heap. [ruby-core:57633] [Bug #8983]
* gc.c (gc_stat): change heap_free_num definition to use new method.
* test/ruby/test_gc.rb: test for above.
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@43139 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 907a03fe - 10/04/2013 10:05 AM - tmm1 (Aman Karmani)

gc.c: add objspace\_free\_num and make GC.stat[:heap\_free\_num] use it

```
* gc.c (objspace_free_num): new method for available/free slots on
heap. [ruby-core:57633] [Bug #8983]
* gc.c (gc_stat): change heap_free_num definition to use new method.
* test/ruby/test_gc.rb: test for above.
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@43139 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

## History

---

### #1 - 10/03/2013 06:29 PM - naruse (Yui NARUSE)

- Status changed from Open to Assigned

- Assignee changed from tmm1 (Aman Karmani) to ko1 (Koichi Sasada)

### #2 - 10/04/2013 07:38 AM - tmm1 (Aman Karmani)

ko1-san, if you agree with this patch I can commit it.


### #3 - 10/04/2013 08:53 AM - ko1 (Koichi Sasada)

(2013/10/03 18:29), naruse (Yui NARUSE) wrote:

<https://github.com/tmm1/ruby/compare/gc-stat-free-num.patch>

One point.


```
@@ -1787,6 +1791,7 @@ struct os_each_struct {
else {
struct heap_slot *slot = (struct heap_slot *) (VALUE)RDATA(p)->dmark;
slot->limit--;
```

- 

```
}
p = tmp;
}
```

Should not reduce heap\_limit here.

```
@@ -2343,6 +2348,7 @@ struct force_finalize_list {
RDATA(pp)->dmark = (void *) (void *) (VALUE)sweep_slot;
pp->as.free.flags /= FL_SINGLETON; /freeing page mark */
}
```

- 

```
sweep_slot->limit = final_num;
unlink_heap_slot(objspace, sweep_slot);

}
```

Should set "heap\_limit -= slot->limit".

This slot is already unavaialble from Ruby level.

--  
// SASADA Koichi at atdot dot net

### #4 - 10/04/2013 10:09 AM - tmm1 (Aman Karmani)

I understand the slot is removed, but objspace\_live\_num includes unfinalized objects so I included them in heap\_limit also.

Otherwise, objspace->heap.limit - objspace\_live\_num(objspace) can become negative.

### #5 - 10/04/2013 10:29 AM - ko1 (Koichi Sasada)

(2013/10/04 10:09), tmm1 (Aman Gupta) wrote:

I understand the slot is removed, but objspace\_live\_num includes unfinalized objects so I included them in heap\_limit also.

Otherwise, objspace->heap.limit - objspace\_live\_num(objspace) can become negative.

I got it.

objspace\_live\_num() should return the count including zombie (waiting to finalization) because they are still living (living dead!!).

However, we can eliminate zombies from live\_num() because they are not living normal objects.

Two approaches?

(1) eliminate zombies from live\_num

(2) count zombie\_num

```
(objspace->heap.limit -  
(objspace_live_num(objspace) - zombie_num)
```

--

// SASADA Koichi at atdot dot net

#### #6 - 10/04/2013 10:53 AM - tmm1 (Aman Karmani)

(2) count zombie\_num

```
(objspace->heap.limit -  
(objspace_live_num(objspace) - zombie_num)
```

I like this approach. This will be useful also for debugging finalizer leaks.

Once we replace heap\_free\_num with this formula, I would like to rename the existing heap.free\_num counter. Do you have any suggestion for a better name? Maybe heap.reused\_num or heap.swept\_num

#### #7 - 10/04/2013 12:44 PM - tmm1 (Aman Karmani)

Actually we already have heap.final\_num. Is this the same as zombie\_num?

#### #8 - 10/04/2013 07:05 PM - tmm1 (Aman Karmani)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r43138.

Aman, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

---

gc.c: add rb\_objspace.limit

```
* gc.c: add rb_objspace.limit to keep accurate count of total heap  
slots \[ruby-core:57633\] [Bug #8983]
```