

Ruby trunk - Bug #5985

miniruby skews "make benchmark" results

02/09/2012 04:05 AM - normalperson (Eric Wong)

Status: Closed									
Priority: Normal									
Assignee: ko1 (Koichi Sasada)									
Target version: 2.0.0									
ruby -v: -	Backport:								
Description									
<p>miniruby has fewer objects and a smaller heap than the normal "ruby", benchmarks that are affected by GC performance is skewed heavily. This is most noticeable with the vm3_gc benchmark using the same Ruby revision/build but different executables (miniruby vs ruby):</p>									
<pre>\$ make \$ make install \$ make benchmark-each ITEM=vm3_gc ruby ../benchmark/driver.rb -v \ --executables="ruby; ./miniruby -I../lib -I. -I.ext/common ../tool/runruby.rb --extout=.ext -- --disable-gems" \ --pattern=vm3_gc --directory=../benchmark total: 1 trial(s) (1 trial(s) for 1 benchmark(s)) 2012-02-08 18:57:12 +0000 target 0: ruby 2.0.0dev (2012-02-08 trunk 34493) [x86_64-linux] target 1: ruby 2.0.0dev (2012-02-08 trunk 34493) [x86_64-linux]</pre>									
<hr/>									
<pre>vm3_gc #!/usr/bin/ruby 5000.times do 100.times do {"xxxx"=>"yyyy"} end end GC.start end ruby 2.0.0dev (2012-02-08 trunk 34493) [x86_64-linux] .2.752270221710205 ruby 2.0.0dev (2012-02-08 trunk 34493) [x86_64-linux] .1.857623815536499</pre>									
<hr/>									
<p>raw data:</p> <pre>[["vm3_gc", [[2.752270221710205], [1.857623815536499]]]]</pre>									
<p>Elapsed time: 4.611220872 (sec)</p>									
<p>benchmark results:</p> <table><thead><tr><th>name</th><th>ruby 2.0.0dev (2012-02-08 trunk 34493) [x86_64-linux]</th><th>ruby 2.0.0dev (2012-02-08 trunk 34493) [x86_64-linux]</th><th>average difference</th></tr></thead><tbody><tr><td>vm3_gc</td><td>2.752</td><td>1.858</td><td>-0.895</td></tr></tbody></table>		name	ruby 2.0.0dev (2012-02-08 trunk 34493) [x86_64-linux]	ruby 2.0.0dev (2012-02-08 trunk 34493) [x86_64-linux]	average difference	vm3_gc	2.752	1.858	-0.895
name	ruby 2.0.0dev (2012-02-08 trunk 34493) [x86_64-linux]	ruby 2.0.0dev (2012-02-08 trunk 34493) [x86_64-linux]	average difference						
vm3_gc	2.752	1.858	-0.895						
<p>average total difference is -0.894646406173706</p>									

History

#1 - 03/11/2012 05:24 PM - ko1 (Koichi Sasada)

- Assignee set to ko1 (Koichi Sasada)

I'll change it.

#2 - 03/18/2012 06:46 PM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Assigned

#3 - 11/26/2012 06:20 PM - ko1 (Koichi Sasada)

- Status changed from Assigned to Closed

I changed to accept multiple -e like "-e path1 -e path2". or "-e label1::path1 -e label2::path2".
You don't need to use ';' character.

#4 - 11/26/2012 06:43 PM - ko1 (Koichi Sasada)

Sorry, the previous comment is for [Bug #7380].

BTW, the current benchmark using ruby' instead ofminiruby'.

The difference is '--disable-gems':

```
$ make benchmark COMPARE_RUBY='installed::~~/tmp/trunk/bin/ruby --disable-gems' OPTS='-p vm3_gc'
```

...

```
benchmark results:  
Execution time (sec)  
name installed built-ruby  
vm3_gc 1.806 1.823
```

I think this ticket can be closed.

(or remove --disable-gems ?)

#5 - 11/27/2012 06:31 AM - normalperson (Eric Wong)

"ko1 (Koichi Sasada)" redmine@ruby-lang.org wrote:

Issue [#5985](#) has been updated by ko1 (Koichi Sasada).

Sorry, the previous comment is for [Bug [#7380](#)].

BTW, the current benchmark using ruby' instead ofminiruby'.

The difference is '--disable-gems':

```
$ make benchmark COMPARE_RUBY='installed::~~/tmp/trunk/bin/ruby --disable-gems' OPTS='-p vm3_gc'
```

...

```
benchmark results:  
Execution time (sec)  
name installed built-ruby  
vm3_gc 1.806 1.823
```

I think this ticket can be closed.

(or remove --disable-gems ?)

I favor removing --disable-gems from the benchmark command.

I think it's unrealistic to use/benchmark Ruby without gems nowadays.

#6 - 01/14/2014 10:46 PM - normalperson (Eric Wong)

Btw, I'm still having this problem and it's very noticeable with:
make benchmark-each ITEM=bm_so_reverse_complement

This is running the same COMPARE_RUBY as what was just installed via
"make install" Strangely, miniruby is the one that is slow, here.

```
$ make benchmark-each ITEM=bm_so_reverse_complement  
ruby 2.2.0dev (2014-01-14 trunk 44595) [x86_64-linux] 1.530159944
```

```
built-ruby      2.93252692
```

```
-----  
raw data:
```

```
[["so_reverse_complement", [[1.530159944], [2.93252692]]]]
```

```
Elapsed time: 4.464871542 (sec)
```

```
-----  
benchmark results:
```

```
Execution time (sec)
```

```
name      ruby 2.2.0dev (2014-01-14 trunk 44595) [x86_64-linux]
```

```
built-ruby
```

```
so_reverse_complement  1.530  2.933
```

```
Speedup ratio: compare with the result of `ruby 2.2.0dev (2014-01-14 trunk 44595) [x86_64-linux]' (greater is better)
```

```
name      built-ruby
```

```
so_reverse_complement  0.522
```

Compare with the following, the numbers only differ in noise:

```
$ ruby --disable=gems ./benchmark/driver.rb -v --executables="a::ruby \  
--disable=gems; b::ruby --disable=gems" \  
--pattern=bm_so_reverse_complement --directory=./benchmark
```

```
a      1.547410982
```

```
b      1.537930862
```

```
-----  
raw data:
```

```
[["so_reverse_complement", [[1.547410982], [1.537930862]]]]
```

```
Elapsed time: 3.087557723 (sec)
```

```
-----  
benchmark results:
```

```
Execution time (sec)
```

```
name      a      b
```

```
so_reverse_complement  1.547  1.538
```

```
Speedup ratio: compare with the result of `a' (greater is better)
```

```
name      b
```

```
so_reverse_complement  1.006
```

#7 - 01/15/2014 01:24 PM - Eregon (Benoit Daloze)

Eric Wong wrote:

ko1 (Koichi Sasada) wrote:

(or remove --disable-gems ?)

I favor removing --disable-gems from the benchmark command.

I think it's unrealistic to use/benchmark Ruby without gems nowadays.

Unfortunately that would make the comparison between different ruby versions more brittle as the gems might change, particularly how much allocations they do, no?

I have found as well some benchmarks in benchmark/ are very sensitive to the heap usage.

#8 - 01/15/2014 01:25 PM - Eregon (Benoit Daloze)

- ruby -v changed from ruby 2.0.0dev (2012-02-08 trunk 34493) [x86_64-linux] to -

Issue [#5985](#) has been updated by Benoit Daloze.

Eric Wong wrote:

ko1 (Koichi Sasada) wrote:

(or remove --disable-gems ?)

I favor removing --disable-gems from the benchmark command.
I think it's unrealistic to use/benchmark Ruby without gems nowadays.

Unfortunately that would make the comparison between different ruby versions more brittle as the gems might change, particularly how much allocations they do, no?

I have found as well some benchmarks in benchmark/ are very sensitive to the heap usage.

Bug #5985: miniruby skews "make benchmark" results
<https://bugs.ruby-lang.org/issues/5985#change-44341>

- Author: Eric Wong
- Status: Closed
- Priority: Low
- Assignee: Koichi Sasada
- Category: build
- Target version: 2.0.0
- ruby -v: ruby 2.0.0dev (2012-02-08 trunk 34493) [x86_64-linux]
- Backport: ----- miniruby has fewer objects and a smaller heap than the normal "ruby", benchmarks that are affected by GC performance is skewed heavily. This is most noticeable with the vm3_gc benchmark using the same Ruby revision/build but different executables (miniruby vs ruby):

```
$ make
$ make install
$ make benchmark-each ITEM=vm3_gc
ruby ../benchmark/driver.rb -v \
--executables="ruby; ./miniruby -I../lib -I../ext/common ../tool/runruby.rb --extout=.ext -- --disable-gems" \
--pattern=vm3_gc --directory=../benchmark
total: 1 trial(s) (1 trial(s) for 1 benchmark(s))
2012-02-08 18:57:12 +0000
target 0: ruby 2.0.0dev (2012-02-08 trunk 34493) [x86_64-linux]
target 1: ruby 2.0.0dev (2012-02-08 trunk 34493) [x86_64-linux]
```

vm3_gc

```
#!/usr/bin/ruby
5000.times do
  100.times do
    {"xxx"=>"yyy"}
  end
  GC.start
end

ruby 2.0.0dev (2012-02-08 trunk 34493) [x86_64-linux] .2.752270221710205
ruby 2.0.0dev (2012-02-08 trunk 34493) [x86_64-linux] .1.857623815536499
```

raw data:

```
[[["vm3_gc", [[2.752270221710205], [1.857623815536499]]]]]
```

Elapased time: 4.611220872 (sec)

benchmark results:

name	ruby 2.0.0dev (2012-02-08 trunk 34493) [x86_64-linux]	ruby 2.0.0dev (2012-02-08 trunk 34493) [x86_64-linux]	average difference
vm3_gc	2.752	1.858	-0.895

average total difference is -0.894646406173706

--

<http://bugs.ruby-lang.org/>

#9 - 01/15/2014 06:38 PM - normalperson (Eric Wong)

eregontp@gmail.com wrote:

Eric Wong wrote:

ko1 (Koichi Sasada) wrote:

(or remove --disable-gems ?)

I favor removing --disable-gems from the benchmark command.
I think it's unrealistic to use/benchmark Ruby without gems nowadays.

Unfortunately that would make the comparison between different ruby versions more brittle as the gems might change, particularly how much allocations they do, no?

Good point. Perhaps keeping --disable-gems globally is better.

I have found as well some benchmarks in benchmark/ are very sensitive to the heap usage.

Exactly.

#10 - 01/19/2014 10:49 AM - normalperson (Eric Wong)

Eric Wong normalperson@yhbt.net wrote:

Btw, I'm still having this problem and it's very noticeable with:
make benchmark-each ITEM=bm_so_reverse_complement

Actually, this benchmark is all over the place because benchmark/driver.rb slurps the gigantic output of this benchmark.

Patch at <https://bugs.ruby-lang.org/issues/9430>

#11 - 01/20/2014 04:02 AM - normalperson (Eric Wong)

Btw, I'm still getting more consistent results with normal ruby instead of miniruby, probably because the encodings still affect heap size and memory layout.

I suggest defining BENCHRUBY in my patch:
<http://bogomips.org/ruby.git/patch?id=1ec0da52b7ce2e0e16f17>

The following changes since commit 0558b8a5c40b93c01f5724fe8a3ab409d4374490:

- ext/psych/lib/psych.rb: fix typo by @jwworth [fix GH-500] * lib/rake/file_list.rb: ditto (2014-01-19 05:48:33 +0000)

are available in the git repository at:

`git://80x24.org/ruby.git benchmark`

for you to fetch changes up to 1ec0da52b7ce2e0e16f1729933ea795b12fea7ae:

`common.mk: define BENCHRUBY to avoid miniruby (2014-01-19 07:02:16 +0000)`

Eric Wong (1):
`common.mk: define BENCHRUBY to avoid miniruby`

`common.mk | 6 +++++-`
`1 file changed, 4 insertions(+), 2 deletions(-)`

#12 - 01/28/2014 08:02 PM - normalperson (Eric Wong)

I would like to commit this soon:

```
* common.mk: define BENCHRUBY to avoid miniruby heap size difference
[ruby-core:59887] [Bug #5985]
```

#13 - 01/30/2014 12:27 PM - ko1 (Koichi Sasada)

```
+BENCHRUBY = ./${PROGRAM} -I$(srcdir)/lib -I. -I$(EXTOUT)/common ${RUN_OPTS}
```

I'm not sure BENCHRUBY works with libruby in builddir (don't use installed libruby?).

#14 - 01/30/2014 06:02 PM - normalperson (Eric Wong)

ko1@atdot.net wrote:

```
+BENCHRUBY = ./$(PROGRAM) -I$(srcdir)/lib -I. -I$(EXTOUT)/common
$(RUN_OPTS)
```

I'm not sure BENCHRUBY works with libruby in builddir (don't use installed libruby?).

Good catch, I tested out-of-tree build, but not --enable-shared.
I can force the use of STATIC_RUBY, it looks like.
I have this working both with and without --enable-shared.

<http://bogomips.org/ruby.git/patch/?id=7495b0fdc7>

```
--- a/common.mk
+++ b/common.mk
@@ -1001,14 +1001,14 @@ COMPARE_RUBY = $(BASERUBY)
ITEM =
OPTS =

-BENCHRUBY = ./$(PROGRAM) -I$(srcdir)/lib -I. -I$(EXTOUT)/common $(RUN_OPTS)
+BENCHRUBY = ./$(STATIC_RUBY) -I$(srcdir)/lib -I. -I$(EXTOUT)/common $(RUN_OPTS)

-benchmark: $(PROGRAM) PHONY
+benchmark: $(PROGRAM) PHONY $(STATIC_RUBY)
$(BASERUBY) $(srcdir)/benchmark/driver.rb -v \
--executables="$(COMPARE_RUBY); built-ruby::$(BENCHRUBY)" \
--pattern='bm_' --directory=$(srcdir)/benchmark $(OPTS)

-benchmark-each: $(PROGRAM) PHONY
+benchmark-each: $(PROGRAM) PHONY $(STATIC_RUBY)
$(BASERUBY) $(srcdir)/benchmark/driver.rb -v \
--executables="$(COMPARE_RUBY); built-ruby::$(BENCHRUBY)" \
--pattern=$(ITEM) --directory=$(srcdir)/benchmark $(OPTS)
```

#15 - 01/31/2014 07:44 AM - ko1 (Koichi Sasada)

```
+BENCHRUBY = ./$(STATIC_RUBY) -I$(srcdir)/lib -I. -I$(EXTOUT)/common $(RUN_OPT
```

We need to discuss which version we want to compare. static_ruby and dynamic (using libruby) ruby show different performance (basically, dynamic ruby is slower).

To avoid such difference, recently I use benchmark/driver.rb directly to compare installed rubies. To do so, I install non-modified trunk and modified trunk before benchmark.

What should happen on "make benchmark"? This rule I wrote is to compare installed "ruby 1.8" and built miniruby. it is not fair comparison but it is enough to check the rough performance.

#16 - 01/31/2014 08:41 AM - normalperson (Eric Wong)

ko1@atdot.net wrote:

```
+BENCHRUBY = ./$(STATIC_RUBY) -I$(srcdir)/lib -I. -I$(EXTOUT)/common $(RUN_OPT
```

We need to discuss which version we want to compare. static_ruby and dynamic (using libruby) ruby show different performance (basically, dynamic ruby is slower).

To avoid such difference, recently I use benchmark/driver.rb directly to compare installed rubies. To do so, I install non-modified trunk and modified trunk before benchmark.

What should happen on "make benchmark"? This rule I wrote is to compare installed "ruby 1.8" and built miniruby. it is not fair comparison but it is enough to check the rough performance.

I can count 3 options right now for "make benchmark":

a) "make benchmark" becomes a help message to tell people to only benchmark with installed Rubies.

b) use LD_PRELOAD for shared Rubies (not portable?)

c) ignore dynamic for benchmark, it will always be slower than static (neither is nearly as fast as miniruby)

I think c) is the easiest path right now, static-ruby is closer to dynamic ruby. miniruby is too far different and unrealistic.

In all cases, we should probably add a warning if any benchmarked ruby is dynamic.

#17 - 02/28/2014 03:10 AM - normalperson (Eric Wong)

May I commit my current STATIC_RUBY version?

It is not perfect for dynamic ruby, but it is better than current "make benchmark". We can make it perfect later.