Ruby - Feature #10165

Use Process.clock_gettime to speed up Benchmark.realtime.

08/23/2014 09:01 PM - phiggins (Pete Higgins)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	

Description

This patch changes the Benchmark.realtime method to use the Process.clock_gettime internally when generating the time elapsed. Calling Process.clock_gettime is faster than the current way of creating Time objects.

I wrote a benchmark script (also attached) to demonstrate the difference:

```
require 'benchmark'

def old_benchmark
  r0 = Time.now
  yield
  Time.now - r0
end

def new_benchmark
  r0 = Process.clock_gettime(Process::CLOCK_MONOTONIC)
  yield
  Process.clock_gettime(Process::CLOCK_MONOTONIC) - r0
end

n = (ARGV.first || 1_000_000).to_i

puts "#{n} iterations."

Benchmark.bmbm do |b|
  b.report("old") { n.times { old_benchmark { nil } } }
  b.report("new") { n.times { new_benchmark { nil } } }
end
```

When I run this on my local machine I see this output:

```
1000000 iterations.
Rehearsal ----------------------------------------
old   0.860000   0.000000   0.860000 (  0.863118)
new   0.360000   0.000000   0.360000 (  0.355242)
------------------------------ total: 1.220000sec

          user     system      total        real
old   0.870000   0.010000   0.880000 (  0.866577)
new   0.330000   0.000000   0.330000 (  0.328982)
```

I discussed this idea originally with Eric Hodel, but he has not reviewed this code.

Associated revisions

Revision 249bd1ed2a00c9c3defdd09224b22a6691f73789 - 08/24/2014 02:03 AM - Eric Wong

lib/benchmark.rb: speedup by reducing allocations

- lib/benchmark.rb (module Benchmark): define BENCHMARK_CLOCK
  (realtime): use Process.clock_gettime(BENCHMARK_CLOCK)
  [Feature #10165]

- test/benchmark/test_benchmark.rb (test_realtime_output): new test

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@47260 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 249bd1ed - 08/24/2014 02:03 AM - Eric Wong

lib/benchmark.rb: speedup by reducing allocations

- lib/benchmark.rb (module Benchmark): define BENCHMARK_CLOCK
  (realtime): use Process.clock_gettime(BENCHMARK_CLOCK)
  [Feature #10165]

- test/benchmark/test_benchmark.rb (test_realtime_output): new test

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@47260 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 08/23/2014 09:48 PM - normalperson (Eric Wong)

I like this. The speedup is from reduction of allocations+GC

I think you need to fall back to CLOCK_REALTIME on systems w/o
CLOCK_MONOTONIC, though. Based on my reading of process.c,
CLOCK_REALTIME is always available. So something like this:

```
if defined?(Process::CLOCK_MONOTONIC)
  BENCHMARK_CLOCK = Process::CLOCK_MONOTONIC
else
  # Ruby may use gettimeofday to emulate:
  BENCHMARK_CLOCK = Process::CLOCK_REALTIME
end

def realtime # :yield:
  r0 = Process.clock_gettime(BENCHMARK_CLOCK)
  yield
  Process.clock_gettime(BENCHMARK_CLOCK) - r0
end
```

#2 - 08/24/2014 12:23 AM - phiggins (Pete Higgins)

- File faster_benchmark_realtime_2.diff added

Eric Wong wrote:

> I like this. The speedup is from reduction of allocations+GC
>
> I think you need to fall back to CLOCK_REALTIME on systems w/o
> CLOCK_MONOTONIC, though. Based on my reading of process.c,
> CLOCK_REALTIME is always available. So something like this:

It wasn't clear from the docs if there was a way to tell which modes were supported. Thanks for looking into that!

I've made an updated patch with the changes Eric Wong suggested.

#3 - 08/24/2014 02:03 AM - Anonymous

- Status changed from Open to Closed

- % Done changed from 0 to 100

Applied in changeset r47260.

---

lib/benchmark.rb: speedup by reducing allocations

- lib/benchmark.rb (module Benchmark): define BENCHMARK_CLOCK
  (realtime): use Process.clock_gettime(BENCHMARK_CLOCK)
  [Feature #10165]

- test/benchmark/test_benchmark.rb (test_realtime_output): new test

#4 - 08/24/2014 02:19 AM - normalperson (Eric Wong)

Thanks Pete! I committed your patch as r47260.
I couldn't resist, so I also made r47622 to modify measure, too.
(smaller improvement, 1.30s vs 1.44s on my VM)

```
require 'benchmark'
p(Benchmark.measure { 100000.times { Benchmark.measure {} } })
```

Files

benchmark_benchmark_realtime.rb	424 Bytes	08/23/2014	phiggins (Pete Higgins)
faster_benchmark_realtime.diff	909 Bytes	08/23/2014	phiggins (Pete Higgins)
faster_benchmark_realtime_2.diff	1.1 KB	08/24/2014	phiggins (Pete Higgins)