Ruby - Bug #3919

Ruby in PowerTOP - too many CPU wakeups

10/08/2010 03:28 PM - sunaku (Suraj Kurapati)

Status:	Closed		
Priority:	Normal		
Assignee:	ko1 (Koichi Sasada)		
Target version:	1.9.3		
ruby -v:	ruby 1.9.2p0 (2010-08-18 revision 29036) [x86_64-linux]	**Backport:**	

Description

=begin
Hello,

Ruby causes too many CPU wakeups according to Intel's PowerTOP tool.
For example, this simple I/O bound program causes 99 wakeups/second
and accounts for 14% of the CPU wakeups on my system:

ruby -e gets

In contrast, when I run python (2.6.5) and make it wait for I/O, it
does not even appear in PowerTOP because it causes so little wakeups.

Please make Ruby more power-efficient! Thanks for your consideration.
=end

Related issues:

Is duplicate of Ruby - Feature #3436: Spawn the timer thread lazily	**Closed**	**06/13/2010**

History

#1 - 10/08/2010 06:08 PM - zenspider (Ryan Davis)

=begin

On Oct 7, 2010, at 23:28 , Suraj Kurapati wrote:

> Ruby causes too many CPU wakeups according to Intel's PowerTOP tool.
> For example, this simple I/O bound program causes 99 wakeups/second
> and accounts for 14% of the CPU wakeups on my system:
>
> ruby -e gets
>
> In contrast, when I run python (2.6.5) and make it wait for I/O, it
> does not even appear in PowerTOP because it causes so little wakeups.
>
> Please make Ruby more power-efficient! Thanks for your consideration.

Ironically, I was just benchmarking perl, python, and ruby for both eval speed and startup speed (which for python is heavily IO bound):

# of iterations = 1000

```
                      user     system      total        real
```

null_time 0.000000 0.000000 0.000000 (0.000121)
perl -e 0 0.040000 0.330000 7.190000 (9.398628)
ruby -e 0 0.030000 0.300000 4.730000 (6.062263)
python -c 0 0.070000 0.450000 74.490000 (77.852470)
perl eval 1 + 1 0.040000 0.390000 10.180000 (11.488141)
ruby eval 1 + 1 0.040000 0.340000 6.770000 (7.036988)
python eval 1 + 1 0.060000 0.440000 75.790000 (77.879552)

It is absolutely amazing to me that python startup costs are so bad.

So, by all means, make ruby more power-efficient, but please do not make it as efficient as python! :P

=end

#2 - 10/08/2010 06:38 PM - Spakman (Mark Somerville)

=begin
On Fri, Oct 08, 2010 at 03:28:26PM +0900, Suraj Kurapati wrote:

> Bug #3919: Ruby in PowerTOP - too many CPU wakeups
> http://redmine.ruby-lang.org/issues/show/3919

This is a duplicate of #3436.

> Ruby causes too many CPU wakeups according to Intel's PowerTOP tool.
> For example, this simple I/O bound program causes 99 wakeups/second
> and accounts for 14% of the CPU wakeups on my system:
>
> ruby -e gets
>
> Please make Ruby more power-efficient! Thanks for your consideration.

This is caused by the timer thread. It's not easy to remove the timer thread completely, however in ruby-core:32686 [1] I submitted a patch to only use the timer thread when it is required to schedule Ruby threads. With that patch, the timer thread is not used in single-threaded applications.

Feedback on my approach to this problem would be greatly appreciated, since this is an important issue for me. The code seems sound, but perhaps there are some intricacies I've missed. Any thoughts?

1. http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-core/32686

Attachment: (unnamed)
=end

#3 - 11/15/2010 06:07 AM - felipebalbi (Felipe Balbi)

=begin
This is actually a regression as it doesn't happen on 1.8.7 (2010-06-23 patchlevel 299).

If I do the same thing with 1.8.7 (ruby -e gets and fire powertop) ruby neve appears on powertop's list.

I would be willing to help looking further into this issue if someone gives me a little hint on how to grab more debugging information out of ruby.
=end

#4 - 03/07/2011 07:23 PM - Spakman (Mark Somerville)

=begin
On Fri, Oct 08, 2010 at 03:28:26PM +0900, Suraj Kurapati wrote:

> Bug #3919: Ruby in PowerTOP - too many CPU wakeups
> http://redmine.ruby-lang.org/issues/show/3919

> Please make Ruby more power-efficient! Thanks for your consideration.

In case you missed it, Koichi posted a potential solution to this problem in an other thread[1]. It would be great to see some more feedback there.

[1] - http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-core/33456
=end

#5 - 06/26/2011 03:48 PM - naruse (Yui NARUSE)

- Status changed from Open to Assigned

- Assignee set to ko1 (Koichi Sasada)

- Target version set to 1.9.3

#6 - 06/29/2011 10:44 PM - kosaki (Motohiro KOSAKI)

- Status changed from Assigned to Closed

Fixed by r32244.

Files

noname	207 Bytes	03/07/2011	Spakman (Mark Somerville)
noname	207 Bytes	04/12/2011	Spakman (Mark Somerville)
noname	207 Bytes	04/12/2011	Spakman (Mark Somerville)