Ruby - Feature #13697

[PATCH]: futex based thread primitives

06/29/2017 03:26 AM - normalperson (Eric Wong)

Status:	Assigned	
Priority:	Normal	
Assignee:	kosaki (Motohiro KOSAKI)	
Target version:		

Description

Assigning to kosaki since he wrote the current GVL.
I'm hoping single-core vm_thread_pass benchmark can be
improved, but I'm not sure...

```
Using bare, Linux-specific futexes instead of relying on
NPTL-provided primitives seems to offer some speedups
in the more realistic benchmarks which release GVL
for IO.

Performance seems stable between multi-core and single-core
benchmarks.  However, there is still more regressions for
single-core systems, but I think it mainly affects esoteric
cases.  Mainly, the io_pipe_rw and vm_thread_pipe benchmarks
are improved across the board, so I am pretty happy
with that.

Some of the performance changes (good or bad) may also
be the result of size reductions between the 40-byte NPTL
mutex and the 4 byte futex shifting data into a different
cache line.

io and thread '-p (_io_|thread)' benchmark results on an
 AMD FX-8320 @ 3.5GHz:

  io_copy_stream_write          1.040
  io_copy_stream_write_socket   1.027
  io_file_create                1.016
  io_file_read                  1.057
  io_file_write                 1.001
  io_nonblock_noex              1.047
  io_nonblock_noex2             1.037
  io_pipe_rw                    1.077
  io_select                     1.024
  io_select2                    1.003
  io_select3                    0.991
  require_thread                8.379
  vm_thread_alive_check1        1.171
  vm_thread_close               1.015
  vm_thread_condvar1            0.979
  vm_thread_condvar2            1.192
  vm_thread_create_join         1.043
  vm_thread_mutex1              0.985
  vm_thread_mutex2              1.005
  vm_thread_mutex3              0.991
  vm_thread_pass                4.563
  vm_thread_pass_flood          0.991
  vm_thread_pipe                1.867
  vm_thread_queue               0.995
  vm_thread_sized_queue         1.050
  vm_thread_sized_queue2        1.079
  vm_thread_sized_queue3        1.073
  vm_thread_sized_queue4        1.087

single core (schedtool -a 0x1 -e ...):
```

```
io_copy_stream_write          1.039
io_copy_stream_write_socket   1.012
io_file_create                1.010
io_file_read                  1.066
io_file_write                 0.999
io_nonblock_noex              1.061
io_nonblock_noex2             1.020
io_pipe_rw                    1.101
io_select                     1.008
io_select2                    1.001
io_select3                    0.992
require_thread                1.005
vm_thread_alive_check1        0.938
vm_thread_close               1.135
vm_thread_condvar1            1.145
vm_thread_condvar2            1.134
vm_thread_create_join         1.146
vm_thread_mutex1              0.999
vm_thread_mutex2              0.999
vm_thread_mutex3              1.001
vm_thread_pass                0.887
vm_thread_pass_flood          0.973
vm_thread_pipe                1.100
vm_thread_queue               1.013
vm_thread_sized_queue         1.125
vm_thread_sized_queue2        1.172
vm_thread_sized_queue3        1.184
vm_thread_sized_queue4        1.081
```

History

#1 - 10/03/2017 10:21 PM - normalperson (Eric Wong)

normalperson@yhbt.net wrote:

> https://bugs.ruby-lang.org/issues/13697
> Assigning to kosaki since he wrote the current GVL.
> I'm hoping single-core vm_thread_pass benchmark can be
> improved, but I'm not sure...

Can anybody else review? I guess kosaki is busy. Thanks.

#2 - 01/28/2018 11:41 PM - normalperson (Eric Wong)

> https://bugs.ruby-lang.org/issues/13697

Note, this may be not as necessary since thread_sync.c stuff
(Mutex/Queue/etc..) no longer use pthread_* primitives
[Feature #13517] [Feature #13552]

... And GVL is a different beast

#3 - 04/03/2024 03:50 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Assigned

Files

0001-thread-futex-based-thread-primitives-another-take.patch	19.1 KB	06/29/2017	normalperson (Eric Wong)