

The Terrible Timeout

by Marc Schwieterman

Overview

- Why Timeout is a problem
- What you can do about it
- Where you may see Timeout

Timeout is Unsafe

Ruby's Thread#raise, Thread#kill, and the timeout.rb standard library based on them are inherently broken and should not be used for any purpose.

— *Charles Nutter, 2008*¹

¹ <http://blog.headius.com/2008/02/ruby-threadraise-threadkill-timeoutrb.html>

Timeout is *Still* Unsafe

In the last 3 months, I've worked with a half dozen Sidekiq users plagued with mysterious stability problems. All were caused by the same thing: Ruby's terrible Timeout module. I strongly urge everyone reading this to remove any usage of Timeout from your codebase.

— *Mike Perham, 2015*²

² <http://www.mikeperham.com/2015/05/08/timeout-rubys-most-dangerous-api/>

Happy Path

```
Timeout.timeout(1) do  
  # do something  
end
```

The Problem³

```
main = Thread.current
timer = Thread.start {
  begin
    sleep sec
    main.raise Timeout::Error
  end
}
return yield
timer.kill
```

³ https://github.com/ruby/ruby/blob/v200_645/lib/timeout.rb

Rescue⁴

```
Timeout.timeout(0.001) do
  begin
    raise
  rescue
    sleep 0.002
    # will code here execute?
  end
end
```

⁴ https://github.com/marcisme/talks/blob/master/TTT/timeout_test.rb

NO

Ensure⁵

```
Timeout.timeout(0.001) do
  begin
    ensure
      sleep 0.002
      # will code here execute?
    end
  end
end
```

⁵ https://github.com/marcisme/talks/blob/master/TTT/timeout_test.rb

NO

Synchronization is Not Atomicity

ActiveRecord::Base.connection⁶

```
def checkout
  synchronize do
    conn = acquire_connection # ConnectionTimeoutError
    conn.lease
    checkout_and_verify(conn) # select 1
  end
end
```

⁶ https://github.com/rails/rails/blob/v4.1.14.2/activerecord/lib/activerecord/connectionadapters/abstract/connection_pool.rb

Thread.handle_interrupt⁷

- `:immediate` - Invoke interrupts immediately
- `:on_blocking` - Invoke interrupts while `BlockingOperation`
- `:never` - Never invoke all interrupts

⁷ http://ruby-doc.org/core-2.0.0/Thread.html#method-c-handle_interrupt

Thread.handle_interrupt⁸

```
Timeout.timeout(0.001) do
  Thread.handle_interrupt(Timeout::Error => :never) do
    begin
      ensure
        sleep 0.002
        # will code here execute?
      end
    end
  end
end
```

⁸ https://github.com/marcisme/talks/blob/master/TTT/timeout_test.rb

Probably. But...

Thread.handle_interrupt

```
Timeout.timeout(0.001) do
  Thread.handle_interrupt(Timeout::Error => :never) do
    begin
      Thread.handle_interrupt(TolerableError => :immediate) do
        # do something that can be safely interrupted
      end
    ensure
      # code here will execute
    end
  end
end
end
```

Where

- Application code - Search for "Timeout.timeout"
- pinglish⁹
- ~~Net::HTTP~~ ???

⁹ <https://github.com/jbarnette/pinglish/blob/b546053/lib/pinglish.rb#L127-L129>

Conclusion

- Timeout is *very* hard to use safely
- You may be using it already; Check your dependencies
- Seek alternative solutions
- Use `Thread.handle_interrupt` if you absolutely must

Marc Schwieterman

- [@mschwieterman](#)
- [marcschwieterman.com](#)
- [github.com/marcisme/talks/tree/master/TTT](#)