

Status:	Closed	
Priority:	Normal	
Assignee:	mame (Yusuke Endoh)	
Target version:		

Description

I'd like to add Coverage.suspend, Coverage.resume, and some methods.

Synopsis

```
1: # target.rb
2: def foo
3:   :foo
4: end
5:
6: def bar
7:   :bar
8: end
9:
10: def baz
11:   :baz
12: end

require "coverage"

# Similar to Coverage.start, but does not start the measurement itself
Coverage.setup(oneshot_lines: true)

load "target.rb"

foo           # This call is not counted
Coverage.resume # Start the measurement
bar           # This call is counted
Coverage.suspend # Stop the measure
baz           # This call is not counted

# The result is only for Line 7, the body of method "bar"
p Coverage.result #=> {"target.rb"=>{:oneshot_lines=>[7]}}
```

Background

The motivation is to divide modules for large web services. For web services with a long history, we tend to lose track of the dependencies between modules. Using this proposal and oneshot coverage, we can gather information about the code used to process a particular endpoint with almost no runtime cost. Gathering the information for some endpoints will give a hint to isolate the modules.

I've received similar requests in the past to make Coverage restartable but I didn't understand the need for it. (Sorry about that!) I heard directly from those who were actually in trouble in our company, and I finally understand. Also, the introduction of oneshot coverage, which can now be measured at almost no cost, has increased the demand for suspendable coverage.

New APIs

- Coverage.setup: Almost the same as Coverage.start but does not start the measurement itself.
- Coverage.resume: Start/resume the coverage measurement.
- Coverage.suspend: Suspend the coverage measurement; it is restartable by using Coverage.resume.
- Coverage.state: Returns the current state: :idle, :suspended, and :running.

Coverage.start(...) is now the same as Coverage.start(...); Coverage.resume.

Coverage.running? is the same as Coverage.state == :running.

Discussion

- Currently, I think Coverage.suspend makes sense only for oneshot coverage, but it supports traditional coverage too, for a unknown use case. However, I may disallow it if we find any problems.
- It is ideal to measure multiple oneshot coverage for each endpoint together, but it was difficult for me to implement it efficiently. My co-workers say that this feature is still valuable even with the limitation.
- Another idea is to use TracePoint. However, I'd like to introduce this feature to the coverage library because (1) the runtime cost of TracePoint seems not to be negligible according to our preliminary experiment, (2) we can use an ecosystem for oneshot coverage (e.g., https://github.com/riseshia/oneshot_coverage), and (3) the changeset for coverage is not so large.

Implementation

<https://github.com/ruby/ruby/pull/4856>

Any comments are welcome.

Associated revisions

Revision 86e3d77abb8a033650937710d1ab009e98647494 - 10/25/2021 11:00 AM - mame (Yusuke Endoh)

Make Coverage suspendable (#4856)

- Make Coverage suspendable

Add Coverage.suspend, Coverage.resume and some methods.

[Feature #18176] [ruby-core:105321]

Revision 86e3d77abb8a033650937710d1ab009e98647494 - 10/25/2021 11:00 AM - mame (Yusuke Endoh)

Make Coverage suspendable (#4856)

- Make Coverage suspendable

Add Coverage.suspend, Coverage.resume and some methods.

[Feature #18176] [ruby-core:105321]

Revision 86e3d77a - 10/25/2021 11:00 AM - mame (Yusuke Endoh)

Make Coverage suspendable (#4856)

- Make Coverage suspendable

Add Coverage.suspend, Coverage.resume and some methods.

[Feature #18176] [ruby-core:105321]

Revision d463b407e6ca059316ed33c39970aa32903667fb - 12/13/2021 04:53 AM - mame (Yusuke Endoh)

NEWS.md: Mention suspendable coverage [Feature #18176]

Revision d463b407e6ca059316ed33c39970aa32903667fb - 12/13/2021 04:53 AM - mame (Yusuke Endoh)

NEWS.md: Mention suspendable coverage [Feature #18176]

Revision d463b407 - 12/13/2021 04:53 AM - mame (Yusuke Endoh)

NEWS.md: Mention suspendable coverage [Feature #18176]

History

#1 - 10/25/2021 08:49 AM - mame (Yusuke Endoh)

- Status changed from Open to Assigned

We discussed this ticket in the dev-meeting, and I accepted this proposal as the author and maintainer of coverage.so.

Currently, coverage.so does not support per-thread coverage measurement, so using Coverage.suspend/resume to gather the information for some endpoints may yield misleading results if it is a multi-thread application. I've already added the caveat to the document.

#2 - 10/25/2021 11:01 AM - mame (Yusuke Endoh)

- *Status changed from Assigned to Closed*

Applied in changeset [git|86e3d77abb8a033650937710d1ab009e98647494](https://github.com/ruby/ruby/commit/86e3d77abb8a033650937710d1ab009e98647494).

Make Coverage suspendable (#4856)

- Make Coverage suspendable

Add Coverage.suspend, Coverage.resume and some methods.

[Feature [#18176](#)] [[ruby-core:105321](#)]