Ruby trunk - Feature #7436

Allow for a "granularity" flag for backtrace_locations

11/26/2012 07:06 AM - sam.saffron (Sam Saffron)

Status:	Assigned	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:		

Description

related to http://bugs.ruby-lang.org/issues/7051

Sometimes one need less information (or more information) associated with backtraces.

It would be nice if one could send in a separate flag informing the VM about the granularity of information required, eg:

caller_locations(0,-current_depth, BacktraceInfo::Label & BacktraceInfo::Lineno)

This allows for one to take quicker backtraces if they need less information, additionally BacktraceInfo::Bindings and BacktraceInfo::Klass could be added which allow you to gather more information for heavy profiling / diagnostics.

History

#1 - 11/26/2012 07:08 AM - sam.saffron (Sam Saffron)

ouch, this was meant to be a feature req not a bug ... cant figure out how to change

#2 - 11/26/2012 07:49 AM - zzak (Zachary Scott)

- Category set to core

- Status changed from Open to Assigned

- Assignee set to ko1 (Koichi Sasada)

- Target version set to 2.0.0

#3 - 11/26/2012 07:49 AM - zzak (Zachary Scott)

- Tracker changed from Bug to Feature

#4 - 11/26/2012 07:54 AM - zzak (Zachary Scott)

- Target version changed from 2.0.0 to 2.6

#5 - 11/26/2012 08:56 AM - ko1 (Koichi Sasada)

- Assignee changed from ko1 (Koichi Sasada) to matz (Yukihiro Matsumoto)

#6 - 11/26/2012 08:59 AM - ko1 (Koichi Sasada)

(2012/11/26 7:06), sam.saffron (Sam Saffron) wrote:

> This allows for one to take quicker backtraces if they need less information, additionally BacktraceInfo::Bindings and BacktraceInfo::Klass could be added which allow you to gather more information for heavy profiling / diagnostics.

I want to reject to get bindings and klasses with this API.

Please ask matz.

--
// SASADA Koichi at atdot dot net

#7 - 12/10/2012 10:20 AM - sam.saffron (Sam Saffron)

matz (Yukihiro Matsumoto) keep in mind, people are already hacking and using this feature externally

https://github.com/banister/binding_of_caller

and

https://github.com/charliesome/better_errors

#8 - 12/10/2012 11:16 AM - headius (Charles Nutter)

As a debugging feature I can support access to arbitrary bindings. As a general, runtime, hot-path feature, there's numerous reasons why it's a terrible idea:

- It exposes all method-local state to *everyone*. Any library anywhere can not only access your local variables but modify them too. A potentially massive security hole.
- It requires that all method bodies everywhere in the system maintain all frame/scope state regardless of whether it's used or not, since "Binding.of_caller" style features could access it at any time.
- It drastically limits optimization potential for Ruby. On a good day, JRuby can be 10x or more faster than MRI largely because we've been able to eliminate framing/scoping overhead. A Binding.of_caller feature would severely damage our performance and very likely limit forever general Ruby performance.

#9 - 12/10/2012 11:29 AM - charliesome (Charlie Somerville)

=begin
Why not introduce a (({$DEBUG})) variable that tells the runtime to enable introspective features like this at the cost of performance? I would love for better_errors to run on JRuby, but at the moment it can't since JRuby leans too far towards the performance side of the argument and there's no way to tell it to disregard performance in favour of debuggability.

=end

#10 - 12/17/2012 06:56 AM - headius (Charles Nutter)

A flag that's enabled at runtime would not really work, since optimizations might already have happened. There needs to be a way to specify that optimizations should be off. In JRuby, the simplest way to do that is to turn off the compiler by passing -X-C to JRuby. In that case, all code will remain interpreted with full bindings available at every ruby level of the stack (we do not provide bindings for native methods). Implementing Binding.of_caller in that scenario would be pretty simple.

JRuby currently has a --debug flag that turns on things like trace functions. It might be reasonable for it to turn on full bindings as well, though it would definitely have a major perf impact.

#11 - 12/25/2017 06:15 PM - naruse (Yui NARUSE)

- Target version deleted (2.6)