Ruby trunk - Feature #6216

SystemStackError backtraces should not be reduced to one line

03/28/2012 04:16 PM - postmodern (Hal Brodigan)

Status:	Closed	
Priority:	Normal	
Assignee:	ko1 (Koichi Sasada)	
Target version:	2.6	

Description

When debugging "SystemStackError: stack level too deep" exceptions, it is not helpful that the backtrace is reduced to one single line. Most of the time Ruby incorrectly identifies where cycles begin, resulting in an unrelated "file:line" as the backtrace. A more useful behaviour would be to print the last 30 lines of the backtrace, and have the developer identify which "file:line" is causing the cycle. This is similar to how JRuby handles SystemStackError backtraces.

Related issues:

Has duplicate Ruby trunk - Feature #9805: Backtrace for SystemStackError	**Closed**	**05/05/2014**

Associated revisions

Revision 053759ed - 06/23/2014 02:35 AM - nobu (Nobuyoshi Nakada)

Backtrace for SystemStackError

- eval.c (setup_exception): set backtrace in system stack error other than the pre-allocated sysstack_error. [Feature #6216]
- proc.c (Init_Proc): freeze the pre-allocated sysstack_error.
- vm_insnhelper.c (vm_stackoverflow): raise new instance for each times without calling any methods to keep the backtrace with no further stack overflow.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@46502 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 46502 - 06/23/2014 02:35 AM - nobu (Nobuyoshi Nakada)

Backtrace for SystemStackError

- eval.c (setup_exception): set backtrace in system stack error other than the pre-allocated sysstack_error. [Feature #6216]
- proc.c (Init_Proc): freeze the pre-allocated sysstack_error.
- vm_insnhelper.c (vm_stackoverflow): raise new instance for each times without calling any methods to keep the backtrace with no further stack overflow.

Revision 46502 - 06/23/2014 02:35 AM - nobu (Nobuyoshi Nakada)

Backtrace for SystemStackError

- eval.c (setup_exception): set backtrace in system stack error other than the pre-allocated sysstack_error. [Feature #6216]
- proc.c (Init_Proc): freeze the pre-allocated sysstack_error.
- vm_insnhelper.c (vm_stackoverflow): raise new instance for each times without calling any methods to keep the backtrace with no further stack overflow.

Revision 46502 - 06/23/2014 02:35 AM - nobu (Nobuyoshi Nakada)

Backtrace for SystemStackError

- eval.c (setup_exception): set backtrace in system stack error other than the pre-allocated sysstack_error. [Feature #6216]
- proc.c (Init_Proc): freeze the pre-allocated sysstack_error.
- vm_insnhelper.c (vm_stackoverflow): raise new instance for each times without calling any methods to keep the backtrace with no further stack overflow.

Revision 46502 - 06/23/2014 02:35 AM - nobu (Nobuyoshi Nakada)

Backtrace for SystemStackError

- eval.c (setup_exception): set backtrace in system stack error other than the pre-allocated sysstack_error. [Feature #6216]
- proc.c (Init_Proc): freeze the pre-allocated sysstack_error.
- vm_insnhelper.c (vm_stackoverflow): raise new instance for each times without calling any methods to keep the backtrace with no further stack overflow.

Revision 46502 - 06/23/2014 02:35 AM - nobu (Nobuyoshi Nakada)

Backtrace for SystemStackError

- eval.c (setup_exception): set backtrace in system stack error other than the pre-allocated sysstack_error. [Feature #6216]
- proc.c (Init_Proc): freeze the pre-allocated sysstack_error.
- vm_insnhelper.c (vm_stackoverflow): raise new instance for each times without calling any methods to keep the backtrace with no further stack overflow.

Revision 46502 - 06/23/2014 02:35 AM - nobu (Nobuyoshi Nakada)

Backtrace for SystemStackError

- eval.c (setup_exception): set backtrace in system stack error other than the pre-allocated sysstack_error. [Feature #6216]
- proc.c (Init_Proc): freeze the pre-allocated sysstack_error.
- vm_insnhelper.c (vm_stackoverflow): raise new instance for each times without calling any methods to keep the backtrace with no further stack overflow.

Revision fd4df3be - 06/28/2014 04:58 AM - nobu (Nobuyoshi Nakada)

eval.c: no overwrite SystemStackError backtrace

- eval.c (setup_exception): should not overwrite SystemStackError backtrace if set already. [ruby-core:63377] [Feature #6216]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@46594 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 46594 - 06/28/2014 04:58 AM - nobu (Nobuyoshi Nakada)

eval.c: no overwrite SystemStackError backtrace

- eval.c (setup_exception): should not overwrite SystemStackError backtrace if set already. [ruby-core:63377] [Feature #6216]

Revision 46594 - 06/28/2014 04:58 AM - nobu (Nobuyoshi Nakada)

eval.c: no overwrite SystemStackError backtrace

- eval.c (setup_exception): should not overwrite SystemStackError backtrace if set already. [ruby-core:63377] [Feature #6216]

Revision 46594 - 06/28/2014 04:58 AM - nobu (Nobuyoshi Nakada)

eval.c: no overwrite SystemStackError backtrace

- eval.c (setup_exception): should not overwrite SystemStackError backtrace if set already. [ruby-core:63377] [Feature #6216]

Revision 46594 - 06/28/2014 04:58 AM - nobu (Nobuyoshi Nakada)

eval.c: no overwrite SystemStackError backtrace

- eval.c (setup_exception): should not overwrite SystemStackError backtrace if set already. [ruby-core:63377] [Feature #6216]

Revision 46594 - 06/28/2014 04:58 AM - nobu (Nobuyoshi Nakada)

eval.c: no overwrite SystemStackError backtrace

- eval.c (setup_exception): should not overwrite SystemStackError backtrace if set already. [ruby-core:63377] [Feature #6216]

Revision 46594 - 06/28/2014 04:58 AM - nobu (Nobuyoshi Nakada)

eval.c: no overwrite SystemStackError backtrace

- eval.c (setup_exception): should not overwrite SystemStackError backtrace if set already. [ruby-core:63377] [Feature #6216]

History

#1 - 03/28/2012 07:27 PM - matz (Yukihiro Matsumoto)

- Status changed from Open to Feedback

Could you be more descriptive, please? Comparing outputs from Ruby and JRuby, for example.

Matz.

#2 - 03/29/2012 10:40 AM - postmodern (Hal Brodigan)

Ruby 1.9.3:

```
/home/hal/.rvm/gems/ruby-1.9.3-p125/gems/dm-core-1.2.0/lib/dm-core/support/equalizer.rb:32: stack level too deep (SystemStackError)
```

JRuby 1.6.6:

```
  SystemStackError: stack level too deep
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:114
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:113
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:153
       resource_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:107
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:191
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:23
                get at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:8
          user_name at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/model/relationship.rb:35
1
             to_ary at /vault/1/code/ronin/ronin/lib/ronin/email_address.rb:259
              Array at org/jruby/RubyKernel.java:327
  target_conditions at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/query.rb:56
       source_scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:89
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:156
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:114
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:113
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:153
       resource_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:107
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:191
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:23
                get at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:8
          user_name at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/model/relationship.rb:35
1
             to_ary at /vault/1/code/ronin/ronin/lib/ronin/email_address.rb:259
              Array at org/jruby/RubyKernel.java:327
  target_conditions at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/query.rb:56
       source_scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:89
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:156
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:114
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:113
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:153
       resource_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:107
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:191
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:23
                get at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:8
          user_name at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/model/relationship.rb:35
1
             to_ary at /vault/1/code/ronin/ronin/lib/ronin/email_address.rb:259
              Array at org/jruby/RubyKernel.java:327
  target_conditions at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/query.rb:56
       source_scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:89
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:156
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:114
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:113
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:153
       resource_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:107
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:191
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
```

```
te/persisted.rb:23
                get at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:8
          user_name at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/model/relationship.rb:35
1
             to_ary at /vault/1/code/ronin/ronin/lib/ronin/email_address.rb:259
              Array at org/jruby/RubyKernel.java:327
  target_conditions at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/query.rb:56
       source_scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:89
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:156
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:114
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:113
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:153
       resource_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:107
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:191
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:23
                get at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:8
          user_name at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/model/relationship.rb:35
1
             to_ary at /vault/1/code/ronin/ronin/lib/ronin/email_address.rb:259
              Array at org/jruby/RubyKernel.java:327
  target_conditions at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/query.rb:56
       source_scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:89
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:156
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:114
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:113
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:153
       resource_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:107
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:191
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:23
                get at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:8
          user_name at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/model/relationship.rb:35
1
             to_ary at /vault/1/code/ronin/ronin/lib/ronin/email_address.rb:259
              Array at org/jruby/RubyKernel.java:327
  target_conditions at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/query.rb:56
       source_scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:89
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:156
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:114
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:113
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:153
       resource_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:107
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:191
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:23
                get at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:8
          user_name at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/model/relationship.rb:35
1
             to_ary at /vault/1/code/ronin/ronin/lib/ronin/email_address.rb:259
              Array at org/jruby/RubyKernel.java:327
  target_conditions at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/query.rb:56
       source_scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:89
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:156
```

```
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:114
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:113
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:153
       resource_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:107
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:191
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:23
                get at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:8
          user_name at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/model/relationship.rb:35
1
             to_ary at /vault/1/code/ronin/ronin/lib/ronin/email_address.rb:259
              Array at org/jruby/RubyKernel.java:327
  target_conditions at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/query.rb:56
       source_scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:89
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:156
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:114
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:113
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:153
       resource_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:107
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:191
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:23
                get at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:8
          user_name at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/model/relationship.rb:35
1
             to_ary at /vault/1/code/ronin/ronin/lib/ronin/email_address.rb:259
              Array at org/jruby/RubyKernel.java:327
  target_conditions at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/query.rb:56
       source_scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:89
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:156
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:114
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:113
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:153
       resource_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:107
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:191
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:23
                get at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:8
          user_name at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/model/relationship.rb:35
1
             to_ary at /vault/1/code/ronin/ronin/lib/ronin/email_address.rb:259
              Array at org/jruby/RubyKernel.java:327
  target_conditions at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/query.rb:56
       source_scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:89
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:156
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:114
              scope at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/repository.rb:113
          query_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/relationshi
p.rb:153
       resource_for at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:107
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/associations/many_to_one
.rb:191
          lazy_load at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:23
                get at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/resource/persistence_sta
te/persisted.rb:8
```

```
          user_name at /home/hal/.rvm/gems/jruby-1.6.6/gems/dm-core-1.2.0/lib/dm-core/model/relationship.rb:35
1
```

Formatting of the method-name, file, line-number aside, JRuby does not attempt to compact repeats within the backtrace.

#3 - 04/07/2012 06:42 AM - postmodern (Hal Brodigan)

If it wasn't clear from the above example, Ruby 1.9.3 returns a backtrace pointing to a line of code in dm-core which has nothing to do with the bug. The actual bug is revealed in the full JRuby backtrace where one can see dm-core calling back into Ronin::EmailAddress#to_ary, causing the infinite recursion.

#4 - 10/27/2012 06:51 AM - ko1 (Koichi Sasada)

- Assignee set to matz (Yukihiro Matsumoto)

- Target version changed from 2.0.0 to 2.6

Matz, could you reply to it?

#5 - 10/31/2012 02:14 AM - matz (Yukihiro Matsumoto)

- Status changed from Feedback to Assigned

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

I don't feel I exactly understand the situation, since the specified line in CRuby backtrace does not appear in JRuby's.
That might mean avoiding backtrace truncation has no meaning, or CRuby could have yet another bug.

Do you have any opinion, ko1?

Matz.

#6 - 11/01/2012 05:40 AM - headius (Charles Nutter)

Some details on the JRuby side of things:

- SystemStackError is not consistently raised. There are a decreasing number of places in JRuby's codebase where we attempt to rescue Java's StackOverflowError and reraise it as SystemStackError. Largely, this is because at the point where we catch a StackOverflow, there may not be sufficient stack available to construct the Ruby exception. So you can't expect to always get SystemStackError in JRuby.
- SystemStackError is an unrecoverable error in any runtime. It can happen at very odd times, including during native bits of MRI, exception handling or ensure blocks, and so on. In general, we treat a stack overflow as a fatal error case.
- JRuby in general will just allow the Java StackOverflowError to propagate as-is. You can catch the Java exception using JRuby's Java integration, but as mentioned above stack errors are generally unrecoverable.
- The JVM (Hotspot, at least) only returns a subset of any deep stack trace, even for non-overflow cases. This is the main reason for the truncated backtrace.

#7 - 04/14/2013 03:36 PM - drkaes (Stefan Kaes)

- File stack-overflow.patch added

It seems to me that a full backtrace can safely be generated (patch against trunk attached).

The one line backtrace behavior was introduced with this patch:
https://github.com/shyouhei/ruby/commit/ecd11fb371b5f4a00d0b0006b325de3c5437b8d2

It avoids the crashing call

```
  rb_funcall(info, rb_intern("backtrace"), 0),
```

which is performed when one calls get_backtrace(mesg), defined in eval_error.c.

On the other hand, rb_make_backtrace() will not grow the stack, as far as I can tell.

I've tested the patch with ruby trunk and 1.9.3-p392.

The reason I came here is that we were bitten by a SystemStackError in production, which we couldn't reproduce, and the one line information wasn't enough to debug the issue.

#8 - 04/18/2013 09:13 PM - drkaes (Stefan Kaes)

It has been pointed out to me that the patched ruby will likely crash when the patched code run on a sigaltstack.

I'm working to sort this out.

#9 - 08/25/2013 12:35 AM - thedarkone (Vit Z)

I'm not sure matz (Yukihiro Matsumoto)/@ko1 understood what postmodern (Hal Brodigan) was trying to describe.

Given an overlfow.rb ruby program:

```
200.times do |i|
  eval <<-RUBY_EVAL, nil, __FILE__, __LINE__ + 1
    def foo_#{i}(&block)
      foo_#{i+1}(&block)
    end
  RUBY_EVAL

  def foo_200
    yield
  end
end

def bar
  foo_0 { bar }
end

bar
```

When run on ruby 2.0, this is the output:

```
vit@localhost ~> ruby -v overflow.rb
ruby 2.0.0p247 (2013-06-27 revision 41674) [x86_64-darwin12.2.0]
overflow.rb:4: stack level too deep (SystemStackError)
```

As can be seen, only a single line of backtrace data is provided, thus the only thing that is known - is the stack was blown while executing a foo_x method, but none of the foo_x methods are recursive (the real culprit is the bar method). Debugging an error like that is nigh impossible on Ruby 2.0.

Contrast this with other VMs:

rbx:

```
vit@localhost ~> ruby -v overflow.rb
rubinius 2.0.0.rc1 (1.9.3 release yyyy-mm-dd JI) [x86_64-apple-darwin12.2.0]
An exception occurred running overflow.rb
    SystemStackError (SystemStackError)

Backtrace:
                    Object#foo_141 at overflow.rb:7
                    Object#foo_140 at overflow.rb:8
                    Object#foo_139 at overflow.rb:8
                          [ snip ... ]
                    Object#foo_1 at overflow.rb:8
                    Object#foo_0 at overflow.rb:8
                      Object#bar at overflow.rb:14
               Object#__script__ at overflow.rb:17
Rubinius::CodeLoader#load_script at kernel/delta/codeloader.rb:68
Rubinius::CodeLoader.load_script at kernel/delta/codeloader.rb:119
         Rubinius::Loader#script at kernel/loader.rb:620
           Rubinius::Loader#main at kernel/loader.rb:821
```

MRI 1.8:

```
vit@localhost ~> ruby -v overflow.rb
ruby 1.8.7 (2012-02-08 MBARI 8/0x6770 on patchlevel 358) [i686-darwin12.2.0], MBARI 0x6770, Ruby Enterprise Ed
ition 2012.02
overflow.rb:4:in `foo_44': stack level too deep (SystemStackError)
    from overflow.rb:4:in `foo_43'
    from overflow.rb:4:in `foo_42'
    from overflow.rb:4:in `foo_41'
    from overflow.rb:4:in `foo_40'
    from overflow.rb:4:in `foo_39'
    from overflow.rb:4:in `foo_38'
    from overflow.rb:4:in `foo_37'
    from overflow.rb:4:in `foo_36'
     ... 5312 levels...
```

```
    from overflow.rb:4:in `foo_1'
    from overflow.rb:4:in `foo_0'
    from overflow.rb:14:in `bar'
    from overflow.rb:17
```

#10 - 05/06/2014 09:32 AM - nobu (Nobuyoshi Nakada)

- Has duplicate Feature #9805: Backtrace for SystemStackError added

#11 - 05/06/2014 09:52 AM - nobu (Nobuyoshi Nakada)

https://github.com/ruby/ruby/pull/608

#12 - 06/23/2014 02:35 AM - nobu (Nobuyoshi Nakada)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

Applied in changeset r46502.

Backtrace for SystemStackError

- eval.c (setup_exception): set backtrace in system stack error other than the pre-allocated sysstack_error. [Feature #6216]
- proc.c (Init_Proc): freeze the pre-allocated sysstack_error.
- vm_insnhelper.c (vm_stackoverflow): raise new instance for each times without calling any methods to keep the backtrace with no further stack overflow.

#13 - 06/27/2014 09:32 PM - ccutrer (Cody Cutrer)

I can confirm that this is working on current master, but with one problem - if the exception is ever rescued and then re-raised (even with just a plain raise, not raise e or anything), the backtrace is reset to where it's re-raised from. I'm forced to comment out the entire rescue block for each one to find the true source

#14 - 06/30/2014 05:47 PM - ccutrer (Cody Cutrer)

... and fixed in my testing today

Files

stack-overflow.patch	630 Bytes	04/14/2013	drkaes (Stefan Kaes)