Ruby master - Bug #5719
Hash::[] can't handle 100000+ args
12/07/2011 11:30 AM - qrush (Nick Quaranto)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>ko1 (Koichi Sasada)</td>
</tr>
<tr>
<td>Target version:</td>
<td>2.6</td>
</tr>
<tr>
<td>ruby -v:</td>
<td>-</td>
</tr>
<tr>
<td>Backport:</td>
<td>2.2: UNKNOWN, 2.3: UNKNOWN, 2.4:</td>
</tr>
<tr>
<td></td>
<td>UNKNOWN</td>
</tr>
</tbody>
</table>

Description
I couldn't pin down an exact size for when/how this occurs, but I have code that was creating hashes just fine with Hash::[] with 300k+ arguments.

```
irb(main):056:0> (0...140000).map { |n| [:a, n] }.tap { |a| Hash[*a] }.size
SystemStackError: stack level too deep
from /Users/qrush/.rbenv/versions/1.9.3-p0/lib/ruby/1.9.1/irb/workspace.rb:80
Maybe IRB bug!
```

```
irb(main):057:0> (0...130000).map { |n| [:a, n] }.tap { |a| Hash[*a] }.size
=> 130000
```

Related issues:
Related to Ruby master - Bug #4040: SystemStackError with Hash[*a] for Large _a_

History

#1 - 12/07/2011 11:47 AM - iterology (John Glass)
I was able to duplicate this issue with the above code in 1.9.2-p290 It didn't matter if I ran it through IRB or from the command line.

#2 - 12/07/2011 12:05 PM - gabrielg (Gabriel Gironda)
It looks to be a variable argument length thing, not a Hash thing in particular.

https://gist.github.com/5037bf83d828c2c119c44

Edit: Not just variable argument length. Ruby just won't have any of your guff if you intend on passing any method a very large number of arguments.

https://gist.github.com/d259449f7bedef92c2e

This is reasonable, I think.

#3 - 12/07/2011 12:23 PM - matz (Yukihiro Matsumoto)
- ruby -v changed from 1.9.3-p0 to -

Hi,

In message "Re: [ruby-core:41511] [ruby-trunk - Bug #5719][Open] Hash::[] can't handle 100000+ args"
on Wed, 7 Dec 2011 11:30:44 +0900, Nick Quaranto nick@quaran.to writes:

[|I couldn't pin down an exact size for when/how this occurs, but I have code that was creating hashes just fine with Hash::[] with 300k+ arguments.

Unfortunately we are living in the very restricted real world. ;-;]

matz.

#4 - 12/07/2011 12:26 PM - deepfryed (Bharanee Rathna)
i shall give you four hints: stack, heap, push, pop

ruby's behavior is expected.

you can pass Hash::[] an array instead i think.

Hash[[[a, 1], [b, 2]]] #=> {a=>1, b=>2}
This is related to [http://redmine.ruby-lang.org/issues/982](http://redmine.ruby-lang.org/issues/982). In terms of what needs to be done, it may even be "the same" bug, although 982 is about a very long array literal, and this is about a very long array created by a splash. At [http://redmine.ruby-lang.org/issues/982](http://redmine.ruby-lang.org/issues/982), Koichi Sasada said 「すみません,1.9.3 の後の課題とさせて下さい.」, i.e. "sorry, but let's deal with this after 1.9.3". So now may be a good time :-).

This is related to [http://redmine.ruby-lang.org/issues/982](http://redmine.ruby-lang.org/issues/982). In terms of what needs to be done, it may even be "the same" bug, although 982 is about a very long array literal, and this is about a very long array created by a splash. At [http://redmine.ruby-lang.org/issues/982](http://redmine.ruby-lang.org/issues/982), Koichi Sasada said 「すみません,1.9.3 の後の課題とさせて下さい.」, i.e. "sorry, but let's deal with this after 1.9.3". So now may be a good time :-).

Thank you. It is correct. And there are no progress on it. Sorry. Should we solve this issue as high priority?

--
// SASADA Koichi at atdot dot net

Hello Koichi,

On 2012/02/25 13:34, SASADA Koichi wrote:

This is related to [http://redmine.ruby-lang.org/issues/982](http://redmine.ruby-lang.org/issues/982). In terms of what needs to be done, it may even be "the same" bug, although 982 is about a very long array literal, and this is about a very long array created by a splash. At [http://redmine.ruby-lang.org/issues/982](http://redmine.ruby-lang.org/issues/982), Koichi Sasada said 「すみません,1.9.3 の後の課題とさせて下さい.」, i.e. "sorry, but let's deal with this after 1.9.3". So now may be a good time :-).

Thank you. It is correct.

Thanks for the confirmation.

And there are no progress on it. Sorry. Should we solve this issue as high priority?

I'm not sure "high priority" is the right word. It's always possible to work around it.

But it's highly annoying when somebody hits this issue. It's also highly counterintuitive: Ruby deals with Arrays of any size automatically, but then can't handle the same size in a literal.

So I very much think that 2.0 is a good point to get rid of this problem. I'm not sure what's involved in fixing it, but if there's something I can contribute, I'll be glad to help.

Regards,  Martin.

---

#9 - 03/18/2012 06:46 PM - shyouhei (Shyouhei Urabe)
- Status changed from Open to Assigned

#10 - 11/26/2012 09:30 AM - ko1 (Koichi Sasada)
- Target version changed from 2.0.0 to 2.6

#11 - 01/31/2017 09:18 AM - ko1 (Koichi Sasada)
- Related to Bug #4040: SystemStackError with Hash[*a] for Large _a_ added
See #4040 and close this ticket.