

Ruby trunk - Bug #8125

lost-tuple bug and fix for Rinda::TupleSpaceProxy.take

03/20/2013 02:21 AM - vjoel (Joel VanderWerf)

Status:	Closed	
Priority:	Normal	
Assignee:	drbrain (Eric Hodel)	
Target version:		
ruby -v:	ruby 2.0.0p0 (2013-02-24) [x86_64-linux]	Backport:
Description		
=begin		
<p>Rinda::TupleSpaceProxy prevents tuple loss during #take by exposing a "port" object on the client that the remote side (the tuplespace server) pushes to, instead of relying on the method return value. Pushing to the port fails if the process that called #take has exited, so the tuple will not be deleted from the tuplespace server.</p>		
<p>However, if the process has not exited, and the thread that called #take was interrupted, the port still exists and accepts push requests (in the main drb thread). In this case the tuple is deleted on the server and not available on the client.</p>		
<p>This is frequently a problem when using irb and manually interrupting take calls. It would also be a problem when using timeouts.</p>		
<p>A concise reproduction of the problem is in the attached thread-int.rb.</p>		
<p>The bug can be fixed by the patch below, which replaces the port array with a custom object that rejects pushes if the call stack has been unwound.</p>		
<p>Note that this patch combines naturally with the faster take patch in #8119.</p>		
<pre>diff --git a/lib/rinda/rinda.rb b/lib/rinda/rinda.rb index 18e284a..057c61a 100644 --- a/lib/rinda/rinda.rb +++ b/lib/rinda/rinda.rb @@ -206,6 +206,23 @@ module Rinda # TupleSpaceProxy allows a remote TupleSpace to appear as local. class TupleSpaceProxy • class Port • attr_reader :val + • def initialize • @open = true • end • def close • @open = false • end + • def push val • raise unless @open • @val = val • nil # so that val doesn't get marshalled again • end • end ## # Creates a new TupleSpaceProxy to wrap +ts+. @@ -222,6 +239,19 @@ module Rinda end ##</pre>		

#8125]

- test/rinda/test_rinda.rb: Test for the above.

Revision 39890 - 03/23/2013 06:34 PM - drbrain (Eric Hodel)

- lib/rinda/rinda.rb: Fixed loss of tuple when remote is alive but the call stack was unwound. Patch by Joel VanderWerf. [ruby-trunk - Bug #8125]
 - test/rinda/test_rinda.rb: Test for the above.

Revision e37f8c0f - 03/24/2013 10:09 PM - drbrain (Eric Hodel)

- lib/rinda/tuplespace.rb: Only return tuple entry once on move, either through port or regular return, not both. This results in a 120% speedup when combined with #8125. Patch by Joel VanderWerf. [ruby-trunk - Feature #8119]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@39923 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 39923 - 03/24/2013 10:09 PM - drbrain (Eric Hodel)

- lib/rinda/tuplespace.rb: Only return tuple entry once on move, either through port or regular return, not both. This results in a 120% speedup when combined with #8125. Patch by Joel VanderWerf. [ruby-trunk - Feature #8119]

Revision 39923 - 03/24/2013 10:09 PM - drbrain (Eric Hodel)

- lib/rinda/tuplespace.rb: Only return tuple entry once on move, either through port or regular return, not both. This results in a 120% speedup when combined with #8125. Patch by Joel VanderWerf. [ruby-trunk - Feature #8119]

Revision 39923 - 03/24/2013 10:09 PM - drbrain (Eric Hodel)

- lib/rinda/tuplespace.rb: Only return tuple entry once on move, either through port or regular return, not both. This results in a 120% speedup when combined with #8125. Patch by Joel VanderWerf. [ruby-trunk - Feature #8119]

Revision 39923 - 03/24/2013 10:09 PM - drbrain (Eric Hodel)

- lib/rinda/tuplespace.rb: Only return tuple entry once on move, either through port or regular return, not both. This results in a 120% speedup when combined with #8125. Patch by Joel VanderWerf. [ruby-trunk - Feature #8119]

Revision 39923 - 03/24/2013 10:09 PM - drbrain (Eric Hodel)

- lib/rinda/tuplespace.rb: Only return tuple entry once on move, either through port or regular return, not both. This results in a 120% speedup when combined with #8125. Patch by Joel VanderWerf. [ruby-trunk - Feature #8119]

Revision 39923 - 03/24/2013 10:09 PM - drbrain (Eric Hodel)

- lib/rinda/tuplespace.rb: Only return tuple entry once on move, either through port or regular return, not both. This results in a 120% speedup when combined with #8125. Patch by Joel VanderWerf. [ruby-trunk - Feature #8119]

History

#1 - 03/20/2013 07:50 AM - hsb (Hiroshi SHIBATA)

- Assignee set to seki (Masatoshi Seki)

#2 - 03/23/2013 04:21 PM - drbrain (Eric Hodel)

- File rinda.rb.8215.patch added

- Status changed from Open to Assigned

I think the safe take should be the default take.

I updated your patch and converted thread-int.rb into a test.

#3 - 03/23/2013 04:26 PM - drbrain (Eric Hodel)

- File rinda.rb.8215.2.patch added

Here is an updated patch that matches the rinda style and removes a leftover p

#4 - 03/23/2013 04:37 PM - drbrain (Eric Hodel)

- File *rinda.rb.8215.3.patch* added

Oops, I forgot to run all the tests, this fixes the remaining tests by restoring an accidentally deleted line.

#5 - 03/23/2013 07:16 PM - seki (Masatoshi Seki)

- Assignee changed from *seki (Masatoshi Seki)* to *drbrain (Eric Hodel)*

please commit it!

#6 - 03/24/2013 03:34 AM - drbrain (Eric Hodel)

- Status changed from *Assigned* to *Closed*

- % Done changed from *0* to *100*

This issue was solved with changeset r39890.
Joel, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

-
- lib/rinda/rinda.rb: Fixed loss of tuple when remote is alive but the call stack was unwound. Patch by Joel VanderWerf. [ruby-trunk - Bug [#8125](#)]
 - test/rinda/test_rinda.rb: Test for the above.

Files

thread-int.rb	1.1 KB	03/20/2013	vjoel (Joel VanderWerf)
rinda.rb.8215.patch	2.84 KB	03/23/2013	drbrain (Eric Hodel)
rinda.rb.8215.2.patch	2.84 KB	03/23/2013	drbrain (Eric Hodel)
rinda.rb.8215.3.patch	2.91 KB	03/23/2013	drbrain (Eric Hodel)