

## Redmine - Defect #11828

### Deadlock when accessing wiki pages

2012-09-13 11:46 - Vincent Mathon

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	High	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Wiki	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Affected version:</b>	2.0.3
<b>Resolution:</b>	Wont fix		

**Description**

**Platform:**  
Redmine 2.0.3  
jRuby 1.6.7, rails 3.2.6  
gem 1.8.24  
Tomcat 6.0.29  
java 1.6.0\_31 (HotSpot) 64-bit under Linux CentOS release 5.5 (Final)

I have built the war file from sources with Warbler 1.3.6 and we don't use specific JVM args for jRuby.

**Description:**  
When two or more users access the same wiki page, http threads become (dead) locked. When all ruby runtimes are locked, Redmine, does not respond any more. The problem is not systematic but frequent and quite easy to reproduce. It seems that wiki pages containing images are more likely to produce the bug. This is a blocking bug since we must restart TOMCAT several times per day.

I have recorded thread dumps to provide you some insights. I am not a ruby developer so I cannot easily interpret them. I have sent also you our web.xml file. We are using org.jruby.rack.RackServlet instead of the filter otherwise we cannot use Redmine integration with Eclipse MyLyn.

Note that everything is OK with Redmine 1.4.4. I have also upgraded to jRuby 1.7.0.preview2 => redmine 2.0.3 still have the problem and Redmine 1.4.4 is still OK.

sincerely  
Vincent

### History

#### #1 - 2012-11-06 16:08 - Vincent Mathon

- File *threaddump-1352213293849.tdump* added

Same with the following stack (redmine becomes useless after 15 minutes since all workers are locked):

Redmine 2.1.2  
jRuby 1.6.7, rails 3.2.8  
gem 1.8.24  
Tomcat 7.0.32  
java 1.7.0\_09 (HotSpot) 32-bit under Windows 7

I do the following tasks for building the war:

- gem install bundler
- gem install warbler
- jRuby - S bundle install --without development test
- rake generate\_secret\_token
- warble config
- warble

One can see this in TOMCAT logs:

```
INFO: INFO: could not acquire application permit within 10.0 seconds (try increasing the pool size)
nov. 06, 2012 3:43:12 PM org.apache.catalina.core.ApplicationContext log
SEVERE: ERROR: application error
org.jruby.rack.AcquireTimeoutException: could not acquire application permit within 10.0 seconds
    at org.jruby.rack.PoolingRackApplicationFactory.acquireApplicationPermit(PoolingRackApplicationFactory.jav
```

```
a:201)
  at org.jruby.rack.PoolingRackApplicationFactory.getApplication(PoolingRackApplicationFactory.java:145)
  at org.jruby.rack.DefaultRackDispatcher.getApplication(DefaultRackDispatcher.java:27)
  at org.jruby.rack.AbstractRackDispatcher.process(AbstractRackDispatcher.java:32)
  at org.jruby.rack.AbstractServlet.service(AbstractServlet.java:37)
  at org.jruby.rack.AbstractServlet.service(AbstractServlet.java:43)
  at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:305)
  at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:210)
  at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:222)
  at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:123)
  at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:472)
  at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:168)
  at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:99)
  at org.apache.catalina.valves.AccessLogValve.invoke(AccessLogValve.java:929)
  at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:118)
  at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:407)
  at org.apache.coyote.http11.AbstractHttp11Processor.process(AbstractHttp11Processor.java:1002)
  at org.apache.coyote.AbstractProtocol$AbstractConnectionHandler.process(AbstractProtocol.java:585)
  at org.apache.tomcat.util.net.AprEndpoint$SocketProcessor.run(AprEndpoint.java:1813)
  at java.util.concurrent.ThreadPoolExecutor.runWorker(Unknown Source)
  at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source)
  at java.lang.Thread.run(Unknown Source)
```

Sincerely  
Vincent MATHON

### #2 - 2013-02-07 16:29 - Vincent Mathon

- Status changed from New to Resolved

Problem solved with Redmine 2.2.2 + JRuby 1.7.2 + TOMCAT 7.0.35

### #3 - 2013-02-07 16:41 - Etienne Massip

- Status changed from Resolved to Closed

- Resolution set to Wont fix

Nice, thanks for the update.

### Files

threaddump-1347357628088.tdump	101 KB	2012-09-13	Vincent Mathon
threaddump-1347358474330.tdump	87.8 KB	2012-09-13	Vincent Mathon
web.xml	979 Bytes	2012-09-13	Vincent Mathon
threaddump-1352213293849.tdump	187 KB	2012-11-06	Vincent Mathon