

Ruby master - Bug #9133

logger rotates log files more than expected

11/21/2013 11:17 AM - no6v (Nobuhiro IMAI)

Status: Closed	
Priority: Normal	
Assignee: nobu (Nobuyoshi Nakada)	
Target version:	
ruby -v: ruby 2.1.0dev (2013-11-18 trunk 43704) [x86_64-linux]	Backport: 1.9.3: UNKNOWN, 2.0.0: UNKNOWN
Description =begin (This may be a same problem as #9130 , but this also happens with 2.0.0 and 1.9.3.) Logger rotates log files more than expected as follow: SH> rm -f log* A>> logger = Logger.new("log", 1, 200) B>> logger = Logger.new("log", 1, 200) SH> ls -r xargs head -v ==> log <== # Logfile created on 2013-11-21 11:02:25 +0900 by logger.rb/31641 A>> 3.times{logger.fatal("FATAL")} SH> ls -r xargs head -v ==> log <== # Logfile created on 2013-11-21 11:02:25 +0900 by logger.rb/31641 F, [2013-11-21T11:02:46.935514 #3164] FATAL -- : FATAL F, [2013-11-21T11:02:46.935700 #3164] FATAL -- : FATAL F, [2013-11-21T11:02:46.935767 #3164] FATAL -- : FATAL A>> logger.debug("DEBUG") # <- trigger for rotation SH> ls -r xargs head -v ==> log.0 <== # Logfile created on 2013-11-21 11:02:25 +0900 by logger.rb/31641 F, [2013-11-21T11:02:46.935514 #3164] FATAL -- : FATAL F, [2013-11-21T11:02:46.935700 #3164] FATAL -- : FATAL F, [2013-11-21T11:02:46.935767 #3164] FATAL -- : FATAL ==> log <== # Logfile created on 2013-11-21 11:03:00 +0900 by logger.rb/31641 D, [2013-11-21T11:03:00.301294 #3164] DEBUG -- : DEBUG B>> logger.debug("DEBUG") # <- also trigger for rotation (unintended) SH> ls -r xargs head -v ==> log.0 <== # Logfile created on 2013-11-21 11:03:00 +0900 by logger.rb/31641 D, [2013-11-21T11:03:00.301294 #3164] DEBUG -- : DEBUG ==> log <== # Logfile created on 2013-11-21 11:03:11 +0900 by logger.rb/31641 D, [2013-11-21T11:03:11.315164 #3169] DEBUG -- : DEBUG In this case, two DEBUG lines displace the three FATAL lines even though enough space is still left on "log" at that time. Logger decides to rotate log files by checking @logdev.dev.stat.size. In case of @logdev.dev already rotated by another process, so logger actually checks the size of "log.0". Trunk can be fixed by a patch in #9130 (#note-1). =end	
Related issues: Is duplicate of Ruby master - Bug #9130: logger continues to write in old log...	Closed 11/20/2013

Associated revisions

Revision 95ed0816 - 12/14/2013 05:43 AM - nobu (Nobuyoshi Nakada)

logger.rb: fix extra log rotation

- lib/logger.rb (lock_shift_log): no need to rotate the log file if it has been rotated by another process. based on the patch by no6v (Nobuhiro IMAI) in [ruby-core:58620]. [Bug #9133]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@44203 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 44203 - 12/14/2013 05:43 AM - nobu (Nobuyoshi Nakada)

logger.rb: fix extra log rotation

- lib/logger.rb (lock_shift_log): no need to rotate the log file if it has been rotated by another process. based on the patch by no6v (Nobuhiro IMAI) in [ruby-core:58620]. [Bug #9133]

Revision 44203 - 12/14/2013 05:43 AM - nobu (Nobuyoshi Nakada)

logger.rb: fix extra log rotation

- lib/logger.rb (lock_shift_log): no need to rotate the log file if it has been rotated by another process. based on the patch by no6v (Nobuhiro IMAI) in [ruby-core:58620]. [Bug #9133]

Revision 44203 - 12/14/2013 05:43 AM - nobu (Nobuyoshi Nakada)

logger.rb: fix extra log rotation

- lib/logger.rb (lock_shift_log): no need to rotate the log file if it has been rotated by another process. based on the patch by no6v (Nobuhiro IMAI) in [ruby-core:58620]. [Bug #9133]

Revision 44203 - 12/14/2013 05:43 AM - nobu (Nobuyoshi Nakada)

logger.rb: fix extra log rotation

- lib/logger.rb (lock_shift_log): no need to rotate the log file if it has been rotated by another process. based on the patch by no6v (Nobuhiro IMAI) in [ruby-core:58620]. [Bug #9133]

Revision 44203 - 12/14/2013 05:43 AM - nobu (Nobuyoshi Nakada)

logger.rb: fix extra log rotation

- lib/logger.rb (lock_shift_log): no need to rotate the log file if it has been rotated by another process. based on the patch by no6v (Nobuhiro IMAI) in [ruby-core:58620]. [Bug #9133]

Revision 44203 - 12/14/2013 05:43 AM - nobu (Nobuyoshi Nakada)

logger.rb: fix extra log rotation

- lib/logger.rb (lock_shift_log): no need to rotate the log file if it has been rotated by another process. based on the patch by no6v (Nobuhiro IMAI) in [ruby-core:58620]. [Bug #9133]

History

#1 - 11/25/2013 09:23 AM - no6v (Nobuhiro IMAI)

After that, the @logdev.dev of process A is "log.0", so [#9133](#) happens. I wrote a test for this issue, also fixes [#9133](#).

```
diff --git a/lib/logger.rb b/lib/logger.rb
index c70b091..071594d 100644
--- a/lib/logger.rb
+++ b/lib/logger.rb
@@ -645,7 +645,7 @@ private
File.open(@filename, File::WRONLY | File::APPEND) do |lock|
lock.flock(File::LOCK_EX) # inter-process locking. will be unlocked at closing file
ino = lock.stat.ino
```

- if ino == File.stat(@filename).ino
- if ino == File.stat(@filename).ino and ino == @dev.stat.ino yield # log shifting else # log shifted by another process (i-node before locking and i-node after locking are different) diff --git a/test/logger/test_logger.rb b/test/logger/test_logger.rb index 04bf779..75da962 100644 --- a/test/logger/test_logger.rb +++ b/test/logger/test_logger.rb @@ -2,6 +2,7 @@ require 'test/unit' require 'logger' require 'tempfile' +require 'tmpdir' require_relative '../ruby/envutil'

```
@@ -542,6 +543,46 @@ class TestLogDevice < Test::Unit::TestCase
end
end
```

- def test_shifting_size_not_rotate_too_much
- d(@filename).send(:add_log_header, @tempfile)
- header_size = @tempfile.size
- message = "" * 99 + "\n"
- shift_size = header_size + message.size * 3 - 1
- opt = {shift_age: 1, shift_size: shift_size} +
- Dir.mktmpdir do |tmpdir|
- begin
- log = File.join(tmpdir, "log")
- logdev1 = d(log, opt)
- logdev2 = d(log, opt) +
- ino = File.stat(log).ino
- assert_equal(ino, logdev1.dev.stat.ino)
- assert_equal(ino, logdev2.dev.stat.ino) +
- 3.times{logdev1.write(message)}
- assert_equal(ino, logdev1.dev.stat.ino)
- assert_equal(ino, logdev2.dev.stat.ino) +
- logdev1.write(message)
- assert_equal(File.stat(log).ino, logdev1.dev.stat.ino)
- assert_equal(ino, logdev2.dev.stat.ino) +
- logdev2.write(message)
- ino = File.stat(log).ino
- assert_equal(ino, logdev1.dev.stat.ino)
- assert_equal(ino, logdev2.dev.stat.ino) +
- logdev1.write(message)
- assert_equal(ino, logdev1.dev.stat.ino)
- assert_equal(ino, logdev2.dev.stat.ino)
- ensure
- logdev1.close if logdev1
- logdev2.close if logdev2
- end
- end
- end + private

```
def run_children(n, args, src)
```

#2 - 11/25/2013 09:24 AM - no6v (Nobuhiro IMAI)

no6v (Nobuhiro IMAI) wrote:

After that, the @logdev.dev of process A is "log.0", so [#9133](#) happens.
I wrote a test for this issue, also fixes [#9133](#).

Both 9133 above are [#9130](#), sorry.

#3 - 11/26/2013 12:05 PM - nobu (Nobuyoshi Nakada)

Probably, you can use File.identical? instead of comparison of ino.

#4 - 11/26/2013 07:23 PM - no6v (Nobuhiro IMAI)

Thanks for your suggestion. I feel this simple condition seems looking good.
I updated a patch using File.identical?.

```
diff --git a/lib/logger.rb b/lib/logger.rb
index c70b091..f1c6cbe 100644
--- a/lib/logger.rb
+++ b/lib/logger.rb
@@ -644,8 +644,7 @@ def lock_shift_log
begin
File.open(@filename, File::WRONLY | File::APPEND) do |lock|
lock.flock(File::LOCK_EX) # inter-process locking. will be unlocked at closing file
```

- ino = lock.stat.ino
- if ino == File.stat(@filename).ino
- if File.identical?(lock, @dev) yield # log shifting else # log shifted by another process (i-node before locking and i-node after locking are different) diff --git a/test/logger/test_logger.rb b/test/logger/test_logger.rb index 04bf779..04fa4cd 100644 --- a/test/logger/test_logger.rb +++ b/test/logger/test_logger.rb @@ -2,6 +2,7 @@ require 'test/unit' require 'logger' require 'tempfile' +require 'tmpdir' require_relative './ruby/envutil'

```
@@ -542,6 +543,44 @@ def test_open_logfile_in_multiprocess
end
end
```

- def test_shifting_size_not_rotate_too_much
- d(@filename).send(:add_log_header, @tempfile)
- header_size = @tempfile.size
- message = "" * 99 + "\n"
- shift_size = header_size + message.size * 3 - 1
- opt = {shift_age: 1, shift_size: shift_size} +
- Dir.mktmpdir do |tmpdir|
- begin
- log = File.join(tmpdir, "log")
- logdev1 = d(log, opt)
- logdev2 = d(log, opt) +
- assert File.identical?(log, logdev1.dev)
- assert File.identical?(log, logdev2.dev) +
- 3.times{logdev1.write(message)}
- assert File.identical?(log, logdev1.dev)
- assert File.identical?(log, logdev2.dev) +
- logdev1.write(message)
- assert File.identical?(log, logdev1.dev)
- assert File.identical?(log + ".0", logdev2.dev) +
- logdev2.write(message)
- assert File.identical?(log, logdev1.dev)
- assert File.identical?(log, logdev2.dev) +
- logdev1.write(message)
- assert File.identical?(log, logdev1.dev)
- assert File.identical?(log, logdev2.dev)
- ensure
- logdev1.close if logdev1
- logdev2.close if logdev2
- end
- end
- end + private

```
def run_children(n, args, src)
```

#5 - 11/26/2013 10:23 PM - nobu (Nobuyoshi Nakada)

(13/11/26 19:23), no6v (Nobuhiro IMAI) wrote:

```
File.open(@filename, File::WRONLY | File::APPEND) do |lock|
  lock.flock(File::LOCK_EX) # inter-process locking. will be unlocked at closing file
```

- ino = lock.stat.ino
- if ino == File.stat(@filename).ino
- if File.identical?(lock, @dev)

No needs to compare with @filename?

#6 - 11/27/2013 12:09 PM - no6v (Nobuhiro IMAI)

No needs to compare with @filename?

```
645 File.open(@filename, File::WRONLY | File::APPEND) do |lock|
646   lock.flock(File::LOCK_EX) # inter-process locking. will be unlocked at closing file
647   if File.identical?(lock, @dev)
```

Nice catch. The race condition is just between L645 and L646.

I had to respect the logic of the original code.

I updated a patch again. Thanks.

```
diff --git a/lib/logger.rb b/lib/logger.rb
index c70b091..3c50cb2 100644
--- a/lib/logger.rb
+++ b/lib/logger.rb
@@ -644,8 +644,7 @@ def lock_shift_log
begin
File.open(@filename, File::WRONLY | File::APPEND) do |lock|
lock.flock(File::LOCK_EX) # inter-process locking. will be unlocked at closing file
```

- ino = lock.stat.ino
- if ino == File.stat(@filename).ino
- if File.identical?(@filename, lock) and File.identical?(lock, @dev) yield # log shifting else # log shifted by another process (i-node before locking and i-node after locking are different) diff --git a/test/logger/test_logger.rb b/test/logger/test_logger.rb index 04bf779..04fa4cd 100644 --- a/test/logger/test_logger.rb +++ b/test/logger/test_logger.rb @@ -2,6 +2,7 @@ require 'test/unit' require 'logger' require 'tempfile' +require 'tmpdir' require_relative '../ruby/envutil'

```
@@ -542,6 +543,44 @@ def test_open_logfile_in_multiprocess
end
end
```

- def test_shifting_size_not_rotate_too_much
- d(@filename).send(:add_log_header, @tempfile)
- header_size = @tempfile.size
- message = "*" * 99 + "\n"
- shift_size = header_size + message.size * 3 - 1
- opt = {shift_age: 1, shift_size: shift_size} +
- Dir.mktmpdir do |tmpdir|
- begin
- log = File.join(tmpdir, "log")
- logdev1 = d(log, opt)
- logdev2 = d(log, opt) +
- assert File.identical?(log, logdev1.dev)
- assert File.identical?(log, logdev2.dev) +
- 3.times{logdev1.write(message)}
- assert File.identical?(log, logdev1.dev)
- assert File.identical?(log, logdev2.dev) +
- logdev1.write(message)
- assert File.identical?(log, logdev1.dev)
- assert File.identical?(log + ".0", logdev2.dev) +
- logdev2.write(message)
- assert File.identical?(log, logdev1.dev)
- assert File.identical?(log, logdev2.dev) +
- logdev1.write(message)
- assert File.identical?(log, logdev1.dev)
- assert File.identical?(log, logdev2.dev)
- ensure
- logdev1.close if logdev1
- logdev2.close if logdev2
- end
- end
- end + private

```
def run_children(n, args, src)
```

#7 - 12/08/2013 08:17 PM - hsb (Hiroshi SHIBATA)

- Assignee set to nobu (Nobuyoshi Nakada)

Can you commit this?

#8 - 12/14/2013 02:43 PM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r44203.
Nobuhiro, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

logger.rb: fix extra log rotation

- lib/logger.rb (lock_shift_log): no need to rotate the log file if it has been rotated by another process. based on the patch by no6v (Nobuhiro IMAI) in [ruby-core:58620]. [Bug #9133]