# Ruby master - Feature #5341

**Add SSL session reuse to Net::HTTP**

09/20/2011 06:55 AM - drbrain (Eric Hodel)

<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>drbrain (Eric Hodel)</td>
</tr>
<tr>
<td>Target version</td>
<td>2.0.0</td>
</tr>
</tbody>
</table>

**Description**

SSL session reuse allows reconnection to an HTTPS server to avoid an SSL handshake which avoids extra computations and network round-trips and increases the performance of SSL connections.

**Associated revisions**

Revision ead728ca - 07/25/2012 12:05 AM - drbrain (Eric Hodel)

- lib/net/http.rb: Added SSL session reuse across connections for a single instance to speed up connection.  [Feature #5341]
- NEWS: ditto
- test/net/http/test_https.rb: Tests for #5341

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@36528 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 36528 - 07/25/2012 12:05 AM - drbrain (Eric Hodel)

- lib/net/http.rb: Added SSL session reuse across connections for a single instance to speed up connection.  [Feature #5341]
- NEWS: ditto
- test/net/http/test_https.rb: Tests for #5341

Revision 36528 - 07/25/2012 12:05 AM - drbrain (Eric Hodel)

- lib/net/http.rb: Added SSL session reuse across connections for a single instance to speed up connection.  [Feature #5341]
- NEWS: ditto
- test/net/http/test_https.rb: Tests for #5341

Revision 36528 - 07/25/2012 12:05 AM - drbrain (Eric Hodel)

- lib/net/http.rb: Added SSL session reuse across connections for a single instance to speed up connection.  [Feature #5341]
- NEWS: ditto
- test/net/http/test_https.rb: Tests for #5341

Revision 36528 - 07/25/2012 12:05 AM - drbrain (Eric Hodel)

- lib/net/http.rb: Added SSL session reuse across connections for a single instance to speed up connection.  [Feature #5341]
- NEWS: ditto
- test/net/http/test_https.rb: Tests for #5341

Revision 36528 - 07/25/2012 12:05 AM - drbrain (Eric Hodel)

- lib/net/http.rb: Added SSL session reuse across connections for a single instance to speed up connection.  [Feature #5341]
- NEWS: ditto
- test/net/http/test_https.rb: Tests for #5341

Revision 36528 - 07/25/2012 12:05 AM - drbrain (Eric Hodel)

- lib/net/http.rb: Added SSL session reuse across connections for a single instance to speed up connection.  [Feature #5341]
- NEWS: ditto
- test/net/http/test_https.rb: Tests for #5341

Revision 36528 - 07/25/2012 12:05 AM - drbrain (Eric Hodel)

- lib/net/http.rb: Added SSL session reuse across connections for a single instance to speed up connection.  [Feature #5341]
- NEWS: ditto
- test/net/http/test_https.rb: Tests for #5341

Revision 36528 - 07/25/2012 12:05 AM - drbrain (Eric Hodel)

- lib/net/http.rb: Added SSL session reuse across connections for a single instance to speed up connection.  [Feature #5341]
- NEWS: ditto
- test/net/http/test_https.rb: Tests for #5341

Revision 36528 - 07/25/2012 12:05 AM - drbrain (Eric Hodel)

- lib/net/http.rb: Added SSL session reuse across connections for a single instance to speed up connection.  [Feature #5341]
- NEWS: ditto
- test/net/http/test_https.rb: Tests for #5341

Revision 36528 - 07/25/2012 12:05 AM - drbrain (Eric Hodel)

- lib/net/http.rb: Added SSL session reuse across connections for a single instance to speed up connection.  [Feature #5341]
- NEWS: ditto
- test/net/http/test_https.rb: Tests for #5341

Revision 36528 - 07/25/2012 12:05 AM - drbrain (Eric Hodel)

- lib/net/http.rb: Added SSL session reuse across connections for a single instance to speed up connection.  [Feature #5341]
- NEWS: ditto
- test/net/http/test_https.rb: Tests for #5341

**History**

#1 - 09/20/2011 07:35 AM - drbrain (Eric Hodel)

- File net.http.rb.ssl_session_reuse.patch added
The original patch has unrelated changes.

#2 - 09/20/2011 07:35 AM - drbrain (Eric Hodel)
- File deleted (net.http.rb.ssl_session_reuse.patch)

#3 - 10/25/2011 06:01 PM - nahi (Hiroshi Nakamura)
The patch does;
- Let HTTP object keeps last SSLSession and reuse it when it tries to connect next time.
- Initializes a SSLContext only at first connection.

Comments;
- Is Net::HTTP object made for reusing? Is ‘5.times { http.start; http.finish }’ supposed to work? (Sorry, I don't know Net::HTTP well...)
- Even so, ssl_parameters is ignored except at first connection, even if some params can be updated after the first connection.
- Reusing SSLSession without checking its timeout might cause compatibility issue though I don't know if that's true or not. We need some compatibility study. Eric, you found an issue with OpenSSL 0.9.7 + Google's false-start SSL server, didn't you?

Adding SSLSession to Net::HTTP is not so bad if it is supposed to work, but I think that making Net::HTTP fat could conflict with your Agent proposal #5064. Isn't it good just adding ssl_session property to Net::HTTP?

#4 - 10/25/2011 07:40 PM - nahi (Hiroshi Nakamura)
According to this tweet, ext/openssl in ruby 1.9.1 or earlier didn't work well with Google's SSL false-start servers, right? https://twitter.com/#!/drbrain/status/115945404830449664

Then it might be a bug that existed in old version of ext/openssl.

#5 - 10/26/2011 11:39 AM - drbrain (Eric Hodel)
=begin
Net::HTTP objects can be reused. You may start and finish a connection as many times as you like (the net-http-persistent gem works this way).

Currently the SSLSession can only be initialized once due to OpenSSL restrictions. To change the values you must make a new Net::HTTP object, so I think changing this behavior (if it is needed) is a separate issue.

$ cat test.rb
require 'socket'
require 'openssl'

key = OpenSSL::PKey::RSA.new 256
cert = OpenSSL::X509::Certificate.new
cert.version = 2
cert.serial = 0
cert.not_before = Time.now
cert.not_after = Time.now + 3600
cert.public_key = key.public_key
cert.subject = OpenSSL::X509::Name.parse 'CN=nobody/DC=example'

store = OpenSSL::X509::Store.new
store.set_default_paths

socket = TCPSocket.new 'localhost', 80 # any open port will do
ssl_context = OpenSSL::SSL::SSLContext.new
ssl_socket = OpenSSL::SSL::SSLSocket.new socket, ssl_context

def try ssl_context, params
  ssl_context.set_params params
rescue
  p params.keys.first => $!.message
end

try ssl_context, @ssl_version => 3
try ssl_context, @key => key
try ssl_context, @cert => cert
try ssl_context, @ca_file => '/nonexistent/file'
try ssl_context, @ca_path => '/nonexistent/path'
try ssl_context, @cert_store => store
try ssl_context, @ciphers => []
try ssl_context, @verify_mode => OpenSSL::SSL::VERIFY_PEER
try ssl_context, @verify_callback => proc { |x| }
try ssl_context, @verify_depth => 2
try ssl_context, :@ssl_timeout => 99

$ make runruby
./miniruby -I./lib -I. -I.ext/common ./tool/runruby.rb --extout=.ext --disable-gems ./test.rb
{:@ssl_version=>"can't modify frozen OpenSSL::SSL::SSLContext"
{:@key=>"can't modify frozen OpenSSL::SSL::SSLContext"
{:@cert=>"can't modify frozen OpenSSL::SSL::SSLContext"
{:@ca_file=>"can't modify frozen OpenSSL::SSL::SSLContext"
{:@ca_path=>"can't modify frozen OpenSSL::SSL::SSLContext"
{:@cert_store=>"can't modify frozen OpenSSL::SSL::SSLContext"
{:@ciphers=>"can't modify frozen OpenSSL::SSL::SSLContext"
{:@verify_mode=>"can't modify frozen OpenSSL::SSL::SSLContext"
{:@verify_callback=>"can't modify frozen OpenSSL::SSL::SSLContext"
{:@verify_depth=>"can't modify frozen OpenSSL::SSL::SSLContext"
{:@ssl_timeout=>"can't modify frozen OpenSSL::SSL::SSLContext"

$ ./miniruby -v
ruby 2.0.0dev (2011-10-25 trunk 33524) [x86_64-darwin11.1.0]

I will update the patch to check for the timeout, I did not know it existed.

With this code:


I was only able to reproduce the issue on Ruby 1.9.1, not Ruby 1.8.7, 1.9.2, 1.9.3 or ruby trunk.

=end

#6 - 10/26/2011 11:49 AM - drbrain (Eric Hodel)
I think it is best to add SSL session reuse to Net::HTTP because it is very hard to add it at a higher layer without adding hooks to Net::HTTP.

Net::HTTP is great at making connections and reading and writing the HTTP protocol. I think a higher layer like in #5064 should stick to implementing the features of HTTP beyond what Net::HTTP is good at.

#7 - 10/26/2011 02:58 PM - MartinBosslet (Martin Bosslet)
Eric Hodel wrote:

I will update the patch to check for the timeout, I did not know it existed.

I could imagine that OpenSSL itself already checks for a timeout. I'm not sure, and we have no related tests for test_ssl_session.rb. I'll look into that, regardless of the outcome I would add specific timeout tests.

#8 - 10/26/2011 10:23 PM - Anonymous
On 10/26/2011 11:39 AM, Eric Hodel wrote:

Net::HTTP objects can be reused. You may start and finish a connection as many times as you like (the net-http-persistent gem works this way).

OK, so it's just I was wrong. I felt that I saw an issue for restarting Net::HTTP with #start after #finish, but I cannot find a ticket. I'm just confusing it with another issue. Either it's already fixed.

Currently the SSLSession can only be initialized once due to OpenSSL restrictions. To change the values you must make a new Net::HTTP object, so I think changing this behavior (if it is needed) is a separate issue.

SSLSocket --<>> SSLContext <<-- Session
<-- A |
| +----------------+
You can't modify SSLContext after you create SSLSocket. You must create new SSLContext for new SSLSocket if you need to update ssl params. It might conflict with semantics of Net::HTTP object, thought I don't have any idea what Net::HTTP object should know.
I will update the patch to check for the timeout, I did not know it existed.

OpenSSL has a client session cache in SSLContext but it's not so useful because you need to keep Sessions by yourself, and pick proper Session for each server. Here's a sample usage of client session cache:

https://github.com/nahi/httpclient/commit/7fc04933961ea3ea5a2aa595172ca7cd29a718f5

You would want to implement session cache instead.

FYI: In contrast to the client session cache, the server session cache is very useful and everyone should use it whenever you need a server session cache.

With this code:


I was only able to reproduce the issue on Ruby 1.9.1, not Ruby 1.8.7, 1.9.2, 1.9.3 or ruby trunk.

Thank you. I'll try to reproduce it.

#9 - 11/10/2011 11:23 AM - drbrain (Eric Hodel)
On Oct 26, 2011, at 6:06 AM, Hiroshi Nakamura wrote:

On 10/26/2011 11:39 AM, Eric Hodel wrote:

Net::HTTP objects can be reused. You may start and finish a connection as many times as you like (the net-http-persistent gem works this way).

OK, so it's just I was wrong. I felt that I saw an issue for restarting Net::HTTP with #start after #finish, but I cannot find a ticket. I'm just confusing it with another issue. Either it's already fixed.

Currently the SSLSession can only be initialized once due to OpenSSL restrictions. To change the values you must make a new Net::HTTP object, so I think changing this behavior (if it is needed) is a separate issue.

SSLSession --<--> SSLContext <--><-- Session
<|> A
| +------------------+

You can't modify SSLContext after you create SSLSocket. You must create new SSLContext for new SSLSocket if you need to update ssl params. It might conflict with semantics of Net::HTTP object, thought I don't have any idea what Net::HTTP object should know.

With net/http in 1.9.3, modifying SSL parameters after Net::HTTP#start has no effect. Adding it as a new feature is not difficult, though (delete two lines). It does not cause any conflicts for Net::HTTP that I can see.

I will update the patch to check for the timeout, I did not know it existed.

OpenSSL has a client session cache in SSLContext but it's not so useful because you need to keep Sessions by yourself, and pick proper Session for each server. Here's a sample usage of client session cache:

https://github.com/nahi/httpclient/commit/7fc04933961ea3ea5a2aa595172ca7cd29a718f5

You would want to implement session cache instead.
I think enabling the session cache is useless for net/http because it is single-connection oriented. Instead, just using an ivar to store the session is OK.

In [http://www.openssl.org/docs/ssl/SSL_CTX_set_session_cache_mode.html](http://www.openssl.org/docs/ssl/SSL_CTX_set_session_cache_mode.html), enabling SSL_SESS_CACHE_CLIENT says:

Client sessions are added to the session cache. As there is no reliable way for the OpenSSL library to know whether a session should be reused or which session to choose (due to the abstract BIO layer the SSL engine does not have details about the connection), the application must select the session to be reused by using the SSL_set_session(3) function. This option is not activated by default.

I think for net/http the client session cache is useless. net/http only connects to one server per instance and will only have one context alive at a time, so the cache will not hold more than one session at a time.

Instead of jumping through the hoops of the client session cache (cache-managing class, callbacks) it will be easier to store the session in an instance variable after connect() and SSL negotiation (since there can only ever be one item in the cache for net/http) and apply the session from the ivar via SSL_set_session (SSLSocket#session)

#10 - 11/17/2011 06:07 PM - MartinBosslet (Martin Bosslet)

Eric Hodel wrote:

On Oct 26, 2011, at 6:06 AM, Hiroshi Nakamura wrote:

On 10/26/2011 11:39 AM, Eric Hodel wrote:

I think enabling the session cache is useless for net/http because it is single-connection oriented. Instead, just using an ivar to store the session is OK.

In [http://www.openssl.org/docs/ssl/SSL_CTX_set_session_cache_mode.html](http://www.openssl.org/docs/ssl/SSL_CTX_set_session_cache_mode.html), enabling SSL_SESS_CACHE_CLIENT says:

Client sessions are added to the session cache. As there is no reliable way for the OpenSSL library to know whether a session should be reused or which session to choose (due to the abstract BIO layer the SSL engine does not have details about the connection), the application must select the session to be reused by using the SSL_set_session(3) function. This option is not activated by default.

I think for net/http the client session cache is useless. net/http only connects to one server per instance and will only have one context alive at a time, so the cache will not hold more than one session at a time.

Instead of jumping through the hoops of the client session cache (cache-managing class, callbacks) it will be easier to store the session in an instance variable after connect() and SSL negotiation (since there can only ever be one item in the cache for net/http) and apply the session from the ivar via SSL_set_session (SSLSocket#session)

We already had discussed some of this on IRC. I looked into the TLS RFCs a couple of days back, and from the discussion and the RFC I conclude the same as Eric, that keeping the session as an instance variable should suffice.

#11 - 11/23/2011 09:28 AM - drbrain (Eric Hodel)

- File net.http.rb.ssl_session_reuse.patch added

Updated patch

#12 - 03/30/2012 08:49 AM - mame (Yusuke Endoh)

- Status changed from Open to Assigned
- Assignee set to naruse (Yui NARUSE)
- Target version changed from 1.9.4 to 2.0.0

I tentatively assign this issue to Naruse-san because he is running for the maintainer of net/http.

--
Yusuke Endoh mame@tsg.ne.jp

#13 - 03/30/2012 11:23 AM - Anonymous

- File noname added

On Fri, Mar 30, 2012 at 08:49:36AM +0900, mame (Yusuke Endoh) wrote:

Issue #5341 has been updated by mame (Yusuke Endoh).
Status changed from Open to Assigned
Assignee set to naruse (Yui NARUSE)
Target version changed from 1.9.4 to 2.0.0

I tentatively assign this issue to Naruse-san because he is running for the maintainer of net/http.

When will elections be held? ;-)

--
Aaron Patterson
http://tenderlovemaking.com/

#14 - 07/18/2012 01:21 PM - naruse (Yui NARUSE)
- Assignee changed from naruse (Yui NARUSE) to drbrain (Eric Hodel)

drbrain (Eric Hodel) wrote:

Updated patch

If Martin checked this patch, I'm ok; commit it.

#15 - 07/19/2012 12:54 AM - MartinBosslet (Martin Bosset)
Eric, nahi and I discussed this a while ago and I think we all agreed finally? I remember I was OK with it :)”

#16 - 07/25/2012 09:06 AM - drbrain (Eric Hodel)
- Status changed from Assigned to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r36528.
Eric, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- lib/net/http.rb: Added SSL session reuse across connections for a single instance to speed up connection. [Feature #5341]
- NEWS: ditto
- test/net/http/test_https.rb: Tests for #5341

#17 - 07/25/2012 10:53 PM - nahi (Hiroshi Nakamura)
Sorry for late reply.
On Thu, Nov 10, 2011 at 11:04 AM, Eric Hodel drbrain@segment7.net wrote:

I will update the patch to check for the timeout, I did not know it existed.

OpenSSL has a client session cache in SSLContext but it's not so useful because you need to keep Sessions by yourself, and pick proper Session for each server. Here's a sample usage of client session cache:

https://github.com/nahi/httpclient/commit/7fc04933961ea3ea5a2aa595172ca7cd29a718f5

You would want to implement session cache instead.

I think enabling the session cache is useless for net/http because it is single-connection oriented. Instead, just using an ivar to store the session is OK.

In http://www.openssl.org/docs/ssl/SSL_CTX_set_session_cache_mode.html, enabling SSL_SESS_CACHE_CLIENT says:

Client sessions are added to the session cache. As there is no reliable way for the OpenSSL library to know whether a session should be reused or which session to choose (due to the abstract BIO layer the SSL engine does not have details about the connection), the application must select the session to be reused by using the SSL_set_session(3) function. This option is not activated by default.

I think for net/http the client session cache is useless. net/http only connects to one server per instance and will only have one context alive at a
time, so the cache will not hold more than one session at a time.

Indeed. That's insufficient.

Instead of jumping through the hoops of the client session cache (cache-managing class, callbacks) it will be easier to store the session in an instance variable after connect() and SSL negotiation (since there can only ever be one item in the cache for net/http) and apply the session from the ivar via SSL_set_session (SSLSocket#session=) when we call connect() again.

I like the new patch. Let's commit it and see how it affects existing servers.

Thank you!

**Files**

<table>
<thead>
<tr>
<th>File Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>net.http.rb.ssl_session_reuse.patch</td>
<td>2.21 KB</td>
<td>09/20/2011</td>
<td>drbrain (Eric Hodel)</td>
</tr>
<tr>
<td>noname</td>
<td>500 Bytes</td>
<td>03/30/2012</td>
<td>Anonymous</td>
</tr>
</tbody>
</table>