Ruby trunk - Feature #4309

[ext/openssl] ASN1 performance enhancement

01/24/2011 09:42 AM - MartinBosslet (Martin Bosslet)

Status:	Closed	
Priority:	Normal	
Assignee:	MartinBosslet (Martin Bosslet)	
Target version:	1.9.3	

Description

=begin
Hi all,

recently I noticed that the method

static int ossl_asn1_default_tag(VALUE obj)

(in ossl_asn1.c) iterates through an internal array each time a default tag is to be looked up resulting in O(n) runtime performance. I thought this to be the ideal situation for using a hash and so I added one with OpenSSL::ASN1Data subclasses acting as keys and the corresponding tags as values. I also did some profiling to see whether the constant lookup time made some impact. I first ran a test encoding and parsing a certificate a couple of times and that's what I got:

Old code:
1.998611 seconds.
2.004065 seconds.
1.981882 seconds.
2.129491 seconds.
1.953846 seconds.
1.957313 seconds.
1.958523 seconds.
1.976004 seconds.
1.925835 seconds.
1.974381 seconds.

New code:
1.886169 seconds.
1.871291 seconds.
1.829164 seconds.
1.927687 seconds.
1.879508 seconds.
1.848399 seconds.
1.942286 seconds.
1.908133 seconds.
1.839384 seconds.
1.861159 seconds.

Not much, but I think the improvement is noticeable. Next I ran a "worst case scenario" for the old code.
I encoded and decoded a Sequence with 4 BMPString values, since BMPString with tag 30 is at the end of the internal array. Here the performance gain was roughly 15%:

Old code worst case:
1.507455 seconds.
1.503871 seconds.
1.563551 seconds.
1.523261 seconds.
1.564125 seconds.
1.526295 seconds.
1.553073 seconds.
1.877483 seconds.
1.543568 seconds.
1.518273 seconds.

New code worst case:
1.347785 seconds.

1.359214 seconds.
1.389248 seconds.
1.466773 seconds.
1.350079 seconds.
1.406290 seconds.
1.393683 seconds.
1.368601 seconds.
1.350600 seconds.
1.373738 seconds

Please find attached the patch (including tests) that would add this improvement to Ruby trunk.

Best regards,
Martin

PS: I also changed the UNIVERSAL_TAG_NAME constant's name for OpenSSL::ASN1::EndOfContent from "EOC" to "END_OF_CONTENT" because all other names are the uppercase versions of their corresponding classes, this would have been the only exception to that rule.
Then I also exported two methods, default_tag and default_tag_of_class, for the ASN1 module. I needed them for something I'm currently working on. These latter changes are independent of the performance improvement described above.
=end

Associated revisions

Revision db874053 - 05/22/2011 12:01 AM - emboss

- ext/openssl/ossl_asn1.c: Default tag lookup in constant time via hash instead of previous linear algorithm. [Ruby 1.9 - Feature #4309][ruby-core:34813]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@31680 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 31680 - 05/22/2011 12:01 AM - emboss

- ext/openssl/ossl_asn1.c: Default tag lookup in constant time via hash instead of previous linear algorithm. [Ruby 1.9 - Feature #4309][ruby-core:34813]

Revision 31680 - 05/22/2011 12:01 AM - emboss

- ext/openssl/ossl_asn1.c: Default tag lookup in constant time via hash instead of previous linear algorithm. [Ruby 1.9 - Feature #4309][ruby-core:34813]

Revision 31680 - 05/22/2011 12:01 AM - emboss

- ext/openssl/ossl_asn1.c: Default tag lookup in constant time via hash instead of previous linear algorithm. [Ruby 1.9 - Feature #4309][ruby-core:34813]

Revision 31680 - 05/22/2011 12:01 AM - emboss

- ext/openssl/ossl_asn1.c: Default tag lookup in constant time via hash instead of previous linear algorithm. [Ruby 1.9 - Feature #4309][ruby-core:34813]

Revision 31680 - 05/22/2011 12:01 AM - emboss

- ext/openssl/ossl_asn1.c: Default tag lookup in constant time via hash instead of previous linear algorithm. [Ruby 1.9 - Feature #4309][ruby-core:34813]

Revision 31680 - 05/22/2011 12:01 AM - emboss

- ext/openssl/ossl_asn1.c: Default tag lookup in constant time via hash instead of previous linear algorithm. [Ruby 1.9 - Feature #4309][ruby-core:34813]

History

#1 - 01/24/2011 10:26 AM - MartinBosslet (Martin Bosslet)

- File class_tag_map2.diff added

=begin
Added tests that further assert behavior of OpenSSL::ASN1::default_tag/default_tag_class_of. Attachment replaces the first one.
=end

#2 - 01/24/2011 11:16 PM - mame (Yusuke Endoh)

=begin
Hi,

2011/1/24 Martin Bosslet redmine@ruby-lang.org:

> recently I noticed that the method
>
> static int ossl_asn1_default_tag(VALUE obj)

Sadly there is no maintainer for openssl currently, so it seems difficult
for your patch to be reviewed certainly.

I reviewed your patch simply, but note that I'm not familiar with openssl.

> Please find attached the patch (including tests) that would add this improvement to Ruby trunk.

CLASS_TAG_MAP is exported and not frozen. A user can modify it.
Is it intended?

> PS: I also changed the UNIVERSAL_TAG_NAME constant's name for OpenSSL::ASN1::EndOfContent from "EOC" to "END_OF_CONTENT" because all other names are the uppercase versions of their corresponding classes, this would have been the only exception to that rule.

One patch should include only one modification. Please do not change
two or more things in one patch.

In addition, I think we should leave "EOC" for compatiblity reason.
It is good to have both "EOC" and "END_OF_CONTENT".

> Then I also exported two methods, default_tag and default_tag_of_class, for the ASN1 module. I needed them for something I'm currently working on. These latter changes are independent of the performance improvement described above.

In principle, please explain a use case when you propose a new feature.
In this case, I will not be able to understand the use case, though :-)

If casual users want to use them, I'm not against it. Otherwise, it is
good to merge it with "something you'are currently working on" together,
I think.

Anyway, thank you for your contribution.

--
Yusuke Endoh mame@tsg.ne.jp

=end

#3 - 01/25/2011 07:37 AM - MartinBosslet (Martin Bosslet)

- File class_tag_map3.diff added

=begin

> Sadly there is no maintainer for openssl currently, so it seems difficult
> for your patch to be reviewed certainly.
>
> I reviewed your patch simply, but note that I'm not familiar with openssl.

Thanks for your time!

> CLASS_TAG_MAP is exported and not frozen. A user can modify it.

> Is it intended?

No, you're right, it should be frozen, thanks for the hint!

> In principle, please explain a use case when you propose a new feature.
> In this case, I will not be able to understand the use case, though :-)
>
> If casual users want to use them, I'm not against it. Otherwise, it is
> good to merge it with "something you'are currently working on" together,
> I think.

You're right, when I read my post again, I noticed that my motivation "came out wrong", I should have given a better one :) I thought about this again and I think that only the "default_tag_of_class" functionality is needed, but probably not by the casual user. Nevertheless, if the functionality would be needed, it is still possible to do a lookup in CLASS_TAG_MAP directly. That's why I agree that it would not be necessary to have a separate method for this feature, but I would still leave CLASS_TAG_MAP exported to provide "default_tag_of_class" implicitly.

> In addition, I think we should leave "EOC" for compatiblity reason.
> It is good to have both "EOC" and "END_OF_CONTENT".

That indeed would be the ideal solution, but unfortunately it's not possible to add redundant values because the index directly relates to the default (universal) class tag, i.e. "EOC" has index 0, "BOOLEAN" 1 etc. So I suppose it's OK to leave "EOC".

I tried to integrate your comments and updated the patch file.

> Anyway, thank you for your contribution.

You're welcome :)

Regards,
Martin

=end

#4 - 05/12/2011 08:16 AM - MartinBosslet (Martin Bosslet)

- Assignee set to MartinBosslet (Martin Bosslet)

#5 - 05/22/2011 09:01 AM - Anonymous

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r31680.
Martin, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- ext/openssl/ossl_asn1.c: Default tag lookup in constant time via hash instead of previous linear algorithm. [Ruby 1.9 - Feature #4309][ruby-core:34813]

#6 - 05/22/2011 09:03 AM - MartinBosslet (Martin Bosslet)

I added only the hash lookup. The hash is not exposed as a Ruby constant, it's just used internally instead.

Regards,
Martin

Files

class_tag_map.diff	7.09 KB	01/24/2011	MartinBosslet (Martin Bosslet)
class_tag_map2.diff	7.59 KB	01/24/2011	MartinBosslet (Martin Bosslet)