Ruby - Feature #16163

Reduce the output of `RubyVM::InstructionSequence#to_binary`

09/11/2019 09:03 AM - NagayamaRyoga (Nagayama Ryoga)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		

Description

Abstract

The output of RubyVM::InstructionSequence#to_binary is extremely large.
We have reduced the output of #to_binary by more than 70%.

The execution speed of RubyVM::InstructionSequence.load_from_binary is about 7% slower, but when reading a binary from a file, it may be faster than the master.

Since Bootsnap gem uses #to_binary, this proposal reduces the compilation cache size of Rails projects to about 1/4.

Background

#to_binary and .load_from_binary are used by Bootsnap gem
that is installed by default in Rails projects since Rails 5.2.
Improving #to_binary output also reduces the compilation cache generated by it.

Implementation

https://github.com/ruby/ruby/pull/2450

Techniques

1. Prevented unnecessary structure fields from being output.

   i.e. MJIT information in struct rb_iseq_constant_body.

2. Output integer value in variable length format such as UTF-8.

```
/*
* Small uint serialization
* 0x00000000_00000000 - 0x00000000_0000007f: 1byte | XXXX XXX1 |
* 0x00000000_00000080 - 0x00000000_00003fff: 2byte | XXXX XX10 | XXXX XXXX |
* 0x00000000_00004000 - 0x00000000_001fffff: 3byte | XXXX X100 | XXXX XXXX | XXXX XXXX |
* 0x00000000_00020000 - 0x00000000_0fffffff: 4byte | XXXX 1000 | XXXX XXXX | XXXX XXXX | XXXX
XXXX |
* ...
* 0x00010000_00000000 - 0x00ffffff_ffffffff: 8byte | 1000 0000 | XXXX XXXX | XXXX XXXX | XXXX
XXXX | XXXX XXXX | XXXX XXXX | XXXX XXXX | XXXX XXXX |
* 0x01000000_00000000 - 0xffffffff_ffffffff: 9byte | 0000 0000 | XXXX XXXX | XXXX XXXX | XXXX
XXXX | XXXX XXXX | XXXX XXXX | XXXX XXXX | XXXX XXXX | XXXX XXXX |
*/
```

3. We integrated ID output mechanism and object serialization.

Evaluation

Environment

OS: Ubuntu 16.04 LTS
CPU: Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz

Memory: 32GB

Simple benchmark

First, We combined the files in the benchmark/ and generated a huge .rb file with 5400 lines.
And We measured the output size of #to_binary and the time taken to load it.

The benchmark code: https://gist.github.com/NagayamaRyoga/d482938f3a03c4556d297bb09c03e1fa

- master (ruby 2.7.0dev (2019-08-17T11:20:04Z master 2a65498ca2) [x86_64-linux])

```
size: 1963764B
                                   user     system      total        real
load_from_binary               4.276000   0.000000   4.276000 (  4.277652)
File.read + load_from_binary   5.060000   0.536000   5.596000 (  5.593620)
```

- This proposal

```
size: 463776B
                                   user     system      total        real
load_from_binary               4.576000   0.004000   4.580000 (  4.580691)
File.read + load_from_binary   4.856000   0.080000   4.936000 (  4.934168)
```

The output size of #to_binary is about 24% (4 times smaller!) of the output of master's.

.load_from_binary is about 7% slower.
However, loading the binary from a file and decoding it (File.read + load_from_binary), it is about 12% faster than master.

A Rails project with Bootsnap

Next, We measured the startup time of the simple Rails project generated with $ rails new.
Bootsnap caches the compilation results at the first boot and uses them to load the application from the next time.

Settings:

```
RAILS_ENV=production
DISABLE_SPRING=1
```

- master
  - Cache (tmp/): 32MB
  - The first boot: Average 1.700s (N=10)
  - Boot from cache: Average 0.588s (N=10)

- proposal
  - Cache (tmp/): 9.4MB
  - The first boot: Average 1.684s (N=10)
  - Boot from cache: Average 0.592s (N=10)

The cache size is now about 30%.
There was no impact on project startup time.

Tests

Passed make test-all with RUBY_ISEQ_DUMP_DEBUG='to_binary'.

```
$ make test-all -j8 RUBY_ISEQ_DUMP_DEBUG=to_binary
../../ruby-dev/revision.h unchanged
Run options:
"--ruby=./miniruby -I../../ruby-dev/lib -I. -I.ext/common  ../../ruby-dev/tool/runruby.rb --extout
=.ext  -- --disable-gems" --excludes-dir=../../ruby-dev/test/excludes --name=!/memory_leak/

# Running tests:
```

```
Finished tests in 46.252333s, 452.6258 tests/s, 57576.1656 assertions/s.
20935 tests, 2663032 assertions, 0 failures, 0 errors, 92 skips

ruby -v: ruby 2.7.0dev (2019-09-05T09:20:11Z alt-bytecode/load_.. 8aa0a1cc4c) [x86_64-linux]
```

Conclusion

The output size of RubyVM::InstructionSequence#to_binary is about 1/4 of the master.
The impact on speed is negligible.
Passed all tests.

Associated revisions

Revision 20baa08d652b844806fab424a2a590408ab613ef - 09/19/2019 08:35 AM - NagayamaRyoga (Nagayama Ryoga)

Improve the output of RubyVM::InstructionSequence#to_binary (#2450)

The output of RubyVM::InstructionSequence#to_binary is extremely large.
We have reduced the output of #to_binary by more than 70%.

The execution speed of RubyVM::InstructionSequence.load_from_binary is about 7% slower, but when reading a binary from a file, it may be faster than the master.

Since Bootsnap gem uses #to_binary, this proposal reduces the compilation cache size of Rails projects to about 1/4.

See details: [Feature #16163]

Revision 20baa08d652b844806fab424a2a590408ab613ef - 09/19/2019 08:35 AM - NagayamaRyoga (Nagayama Ryoga)

Improve the output of RubyVM::InstructionSequence#to_binary (#2450)

The output of RubyVM::InstructionSequence#to_binary is extremely large.
We have reduced the output of #to_binary by more than 70%.

The execution speed of RubyVM::InstructionSequence.load_from_binary is about 7% slower, but when reading a binary from a file, it may be faster than the master.

Since Bootsnap gem uses #to_binary, this proposal reduces the compilation cache size of Rails projects to about 1/4.

See details: [Feature #16163]

Revision 20baa08d - 09/19/2019 08:35 AM - NagayamaRyoga (Nagayama Ryoga)

Improve the output of RubyVM::InstructionSequence#to_binary (#2450)

The output of RubyVM::InstructionSequence#to_binary is extremely large.
We have reduced the output of #to_binary by more than 70%.

The execution speed of RubyVM::InstructionSequence.load_from_binary is about 7% slower, but when reading a binary from a file, it may be faster than the master.

Since Bootsnap gem uses #to_binary, this proposal reduces the compilation cache size of Rails projects to about 1/4.

See details: [Feature #16163]

Revision 644336eef54c8ee2aeb7fd6c55fcd5620bcfa5b4 - 12/21/2019 08:20 PM - ko1 (Koichi Sasada)

add a NEWS entry for [Feature #16163]

Revision 644336eef54c8ee2aeb7fd6c55fcd5620bcfa5b4 - 12/21/2019 08:20 PM - ko1 (Koichi Sasada)

add a NEWS entry for [Feature #16163]

Revision 644336ee - 12/21/2019 08:20 PM - ko1 (Koichi Sasada)

add a NEWS entry for [Feature #16163]

History

#1 - 09/19/2019 08:35 AM - NagayamaRyoga (Nagayama Ryoga)

- Status changed from Open to Closed

Applied in changeset git|20baa08d652b844806fab424a2a590408ab613ef.

Improve the output of RubyVM::InstructionSequence#to_binary (#2450)

The output of RubyVM::InstructionSequence#to_binary is extremely large.
We have reduced the output of #to_binary by more than 70%.

The execution speed of RubyVM::InstructionSequence.load_from_binary is about 7% slower, but when reading a binary from a file, it may be faster than the master.

Since Bootsnap gem uses #to_binary, this proposal reduces the compilation cache size of Rails projects to about 1/4.

See details: [Feature #16163]