Ruby trunk - Feature #8509

Use 128 bit integer type in Bignum

06/10/2013 09:59 PM - akr (Akira Tanaka)

Status:	Closed
Priority:	Normal
Assignee:	akr (Akira Tanaka)
Target version:	2.1.0

Description

How about Bignum uses 128 bit integer type?

I found that recent gcc (since gcc 4.6) supports 128 bit integer type,
__int128, on some platforms.
http://gcc.gnu.org/gcc-4.6/changes.html

It seems gcc supports it on x86_64 and not on i386.

Currently Ruby implements Bignum on top of 32 bit integer type (BDIGIT)
and 64 bit integer type (BDIGIT_DBL).
(Ruby uses two integer types for multiplication.
BDIGIT_DBL can represent any value of BDIGIT * BDIGIT.)

Historically, Ruby supported platforms without 64 bit integer type.
Ruby used 16 bit integer type (BDIGIT) and 32 bit integer type (BDIGIT_DBL)
on such platform.
However I guess no one use such platforms today.

So with gcc 4.6 or later, we can use 64 bit integer type (BDIGIT) and
128 bit integer type (BDIGIT_DBL).

This may gain performance.

I implemented it. (int128-bignum.patch)

Simple benchmark on Debian GNU/Linux 7.0 (wheezy) x86_64:

trunk% time ./ruby -e 'v = 3*1000; u = 1; 1000.times { u *= v }'
./ruby -e 'v = 31000; u = 1; 1000.times { u *= v }' 1.64s user 0.00s system 99% cpu 1.655 total
128bit% time ./ruby -e 'v = 31000; u = 1; 1000.times { u *= v }'
./ruby -e 'v = 3*1000; u = 1; 1000.times { u *= v }' 1.21s user 0.01s system 99% cpu 1.222 total

I think larger integer type reduces control overhead and compiler will have more opportunity for optimization.

However the patch has API incompatibility.

BDIGIT and BDIGIT_DBL and related definitions are defined in a public headers,
ruby/defines.h.

So third party extensions may be broken with the change.

Note that BDIGIT_DBL is a macro (not typedef name), compiler used for third party extension
don't need to support __int128 unless the extension actually uses BDIGIT_DBL.

If a program try to extract information from a Bignum and assumes BDIGIT is 32 bit integer,
the result may be invalid.
In this situation rb_big_pack/rb_big_unpack or rb_integer_pack/rb_integer_unpack [ruby-core:55408] may help.

However BDIGIT size change itself may cause problems.

One example I patched is about rb_big_pow.
int128-bignum.patch contains following modification for rb_big_pow.

- const long BIGLEN_LIMIT = BITSPERDIG*1024*1024;

- const long BIGLEN_LIMIT = 32*1024*1024;

BIGLEN_LIMIT controls the rb_big_pow generates a Bignum or a Float.
If it is not modified, a test causes memory allocation failure.

Another problem is bigdecimal tests.
bigdecimal tests failed with int128-bignum.patch as follows.

1) Failure:
TestBigDecimal#test_power_of_three [/home/akr/tst1/ruby/test/bigdecimal/test_bigdecimal.rb:1006]:
<(1/81)> expected but was
<#>.

2) Failure:
TestBigDecimal#test_power_with_prec [/home/akr/tst1/ruby/test/bigdecimal/test_bigdecimal.rb:1110]:
<#> expected but was
<#>.

3) Failure:
TestBigDecimal#test_power_without_prec [/home/akr/tst1/ruby/test/bigdecimal/test_bigdecimal.rb:1103]:
<#> expected but was
<#>.

4) Failure:
TestBigDecimal#test_sqrt_bigdecimal [/home/akr/tst1/ruby/test/bigdecimal/test_bigdecimal.rb:796]:
expected but was
<#>.

5) Failure:
TestBigMath#test_atan [/home/akr/tst1/ruby/test/bigdecimal/test_bigmath.rb:60]:
[ruby-dev:41257].
<#> expected but was
<#>.

I guess bigdecimal determines precision depend on sizeof(BDIGIT).
I think it is not a good way to use BDIGIT.

How do you think, mrkn?

Also, we cannot define PRI_BDIGIT_DBL_PREFIX because
there is no printf directive for __int128.

Anyway, is Bignum with __int128 worth to support?
Any opinion?

Associated revisions

Revision 1a6a65f1 - 06/18/2013 09:46 AM - akr (Akira Tanaka)

- configure.in: Check __int128.

- include/ruby/defines.h (BDIGIT_DBL): Use uint128_t if it is available.
  (BDIGIT): Use uint64_t if uint128_t is available.
  (SIZEOF_BDIGITS): Defined for above case.
  (BDIGIT_DBL_SIGNED): Ditto.
  (PRI_BDIGIT_PREFIX): Ditto.

- include/ruby/ruby.h (PRI_64_PREFIX): Defined.

- bignum.c (rb_big_pow): Don't use BITSPERDIG for the condition which
  rb_big_pow returns Float or Bignum.

[ruby-dev:47413] [Feature #8509]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@41379 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 41379 - 06/18/2013 09:46 AM - akr (Akira Tanaka)

- configure.in: Check __int128.
- include/ruby/defines.h (BDIGIT_DBL): Use uint128_t if it is available.
  (BDIGIT): Use uint64_t if uint128_t is available.
  (SIZEOF_BDIGITS): Defined for above case.
  (BDIGIT_DBL_SIGNED): Ditto.
  (PRI_BDIGIT_PREFIX): Ditto.
- include/ruby/ruby.h (PRI_64_PREFIX): Defined.
- bignum.c (rb_big_pow): Don't use BITSPERDIG for the condition which rb_big_pow returns Float or Bignum.

[ruby-dev:47413] [Feature #8509]

Revision 41379 - 06/18/2013 09:46 AM - akr (Akira Tanaka)

- configure.in: Check __int128.
- include/ruby/defines.h (BDIGIT_DBL): Use uint128_t if it is available.
  (BDIGIT): Use uint64_t if uint128_t is available.
  (SIZEOF_BDIGITS): Defined for above case.
  (BDIGIT_DBL_SIGNED): Ditto.
  (PRI_BDIGIT_PREFIX): Ditto.
- include/ruby/ruby.h (PRI_64_PREFIX): Defined.
- bignum.c (rb_big_pow): Don't use BITSPERDIG for the condition which rb_big_pow returns Float or Bignum.

[ruby-dev:47413] [Feature #8509]

Revision 41379 - 06/18/2013 09:46 AM - akr (Akira Tanaka)

- configure.in: Check __int128.
- include/ruby/defines.h (BDIGIT_DBL): Use uint128_t if it is available.
  (BDIGIT): Use uint64_t if uint128_t is available.
  (SIZEOF_BDIGITS): Defined for above case.
  (BDIGIT_DBL_SIGNED): Ditto.
  (PRI_BDIGIT_PREFIX): Ditto.
- include/ruby/ruby.h (PRI_64_PREFIX): Defined.
- bignum.c (rb_big_pow): Don't use BITSPERDIG for the condition which rb_big_pow returns Float or Bignum.

[ruby-dev:47413] [Feature #8509]

Revision 41379 - 06/18/2013 09:46 AM - akr (Akira Tanaka)

- configure.in: Check __int128.
- include/ruby/defines.h (BDIGIT_DBL): Use uint128_t if it is available.
  (BDIGIT): Use uint64_t if uint128_t is available.
  (SIZEOF_BDIGITS): Defined for above case.
  (BDIGIT_DBL_SIGNED): Ditto.
  (PRI_BDIGIT_PREFIX): Ditto.

- include/ruby/ruby.h (PRI_64_PREFIX): Defined.
- bignum.c (rb_big_pow): Don't use BITSPERDIG for the condition which rb_big_pow returns Float or Bignum.

[ruby-dev:47413] [Feature #8509]

Revision 41379 - 06/18/2013 09:46 AM - akr (Akira Tanaka)

- configure.in: Check __int128.
- include/ruby/defines.h (BDIGIT_DBL): Use uint128_t if it is available.
  (BDIGIT): Use uint64_t if uint128_t is available.
  (SIZEOF_BDIGITS): Defined for above case.
  (BDIGIT_DBL_SIGNED): Ditto.
  (PRI_BDIGIT_PREFIX): Ditto.
- include/ruby/ruby.h (PRI_64_PREFIX): Defined.
- bignum.c (rb_big_pow): Don't use BITSPERDIG for the condition which rb_big_pow returns Float or Bignum.

[ruby-dev:47413] [Feature #8509]

Revision 41379 - 06/18/2013 09:46 AM - akr (Akira Tanaka)

- configure.in: Check __int128.
- include/ruby/defines.h (BDIGIT_DBL): Use uint128_t if it is available.
  (BDIGIT): Use uint64_t if uint128_t is available.
  (SIZEOF_BDIGITS): Defined for above case.
  (BDIGIT_DBL_SIGNED): Ditto.
  (PRI_BDIGIT_PREFIX): Ditto.
- include/ruby/ruby.h (PRI_64_PREFIX): Defined.
- bignum.c (rb_big_pow): Don't use BITSPERDIG for the condition which rb_big_pow returns Float or Bignum.

[ruby-dev:47413] [Feature #8509]

Revision a461f2f8 - 06/20/2013 03:42 PM - akr (Akira Tanaka)

- ext/bigdecimal: Workaround fix for bigdecimal test failures caused by [ruby-dev:47413] [Feature #8509]
- ext/bigdecimal/bigdecimal.h (BDIGIT): Make it independent from the definition for bignum.c.
  (SIZEOF_BDIGITS): Ditto.
  (BDIGIT_DBL): Ditto.
  (BDIGIT_DBL_SIGNED): Ditto.
  (PRI_BDIGIT_PREFIX): Undefine the definition.
  (PRI_BDIGIT_DBL_PREFIX): Ditto.
- ext/bigdecimal/bigdecimal.c (RBIGNUM_ZERO_P): Use rb_bigzero_p.
  (bigzero_p): Removed.
  (is_even): Use rb_big_pack.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@41502 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 41502 - 06/20/2013 03:42 PM - akr (Akira Tanaka)

- ext/bigdecimal: Workaround fix for bigdecimal test failures caused by [ruby-dev:47413] [Feature #8509]

- ext/bigdecimal/bigdecimal.h (BDIGIT): Make it independent from the definition for bignum.c.
  (SIZEOF_BDIGITS): Ditto.
  (BDIGIT_DBL): Ditto.
  (BDIGIT_DBL_SIGNED): Ditto.
  (PRI_BDIGIT_PREFIX): Undefine the definition.
  (PRI_BDIGIT_DBL_PREFIX): Ditto.

- ext/bigdecimal/bigdecimal.c (RBIGNUM_ZERO_P): Use rb_bigzero_p.
  (bigzero_p): Removed.
  (is_even): Use rb_big_pack.

Revision 41502 - 06/20/2013 03:42 PM - akr (Akira Tanaka)

- ext/bigdecimal: Workaround fix for bigdecimal test failures caused by [ruby-dev:47413] [Feature #8509]

- ext/bigdecimal/bigdecimal.h (BDIGIT): Make it independent from the definition for bignum.c.
  (SIZEOF_BDIGITS): Ditto.
  (BDIGIT_DBL): Ditto.
  (BDIGIT_DBL_SIGNED): Ditto.
  (PRI_BDIGIT_PREFIX): Undefine the definition.
  (PRI_BDIGIT_DBL_PREFIX): Ditto.

- ext/bigdecimal/bigdecimal.c (RBIGNUM_ZERO_P): Use rb_bigzero_p.
  (bigzero_p): Removed.
  (is_even): Use rb_big_pack.

Revision 41502 - 06/20/2013 03:42 PM - akr (Akira Tanaka)

- ext/bigdecimal: Workaround fix for bigdecimal test failures caused by [ruby-dev:47413] [Feature #8509]

- ext/bigdecimal/bigdecimal.h (BDIGIT): Make it independent from the definition for bignum.c.
  (SIZEOF_BDIGITS): Ditto.
  (BDIGIT_DBL): Ditto.
  (BDIGIT_DBL_SIGNED): Ditto.
  (PRI_BDIGIT_PREFIX): Undefine the definition.
  (PRI_BDIGIT_DBL_PREFIX): Ditto.

- ext/bigdecimal/bigdecimal.c (RBIGNUM_ZERO_P): Use rb_bigzero_p.
  (bigzero_p): Removed.
  (is_even): Use rb_big_pack.

Revision 41502 - 06/20/2013 03:42 PM - akr (Akira Tanaka)

- ext/bigdecimal: Workaround fix for bigdecimal test failures caused by [ruby-dev:47413] [Feature #8509]

- ext/bigdecimal/bigdecimal.h (BDIGIT): Make it independent from the definition for bignum.c.
  (SIZEOF_BDIGITS): Ditto.

(BDIGIT_DBL): Ditto.
(BDIGIT_DBL_SIGNED): Ditto.
(PRI_BDIGIT_PREFIX): Undefine the definition.
(PRI_BDIGIT_DBL_PREFIX): Ditto.

- ext/bigdecimal/bigdecimal.c (RBIGNUM_ZERO_P): Use rb_bigzero_p.
(bigzero_p): Removed.
(is_even): Use rb_big_pack.

Revision 41502 - 06/20/2013 03:42 PM - akr (Akira Tanaka)

- ext/bigdecimal: Workaround fix for bigdecimal test failures caused by [ruby-dev:47413] [Feature #8509]

- ext/bigdecimal/bigdecimal.h (BDIGIT): Make it independent from the definition for bignum.c.
(SIZEOF_BDIGITS): Ditto.
(BDIGIT_DBL): Ditto.
(BDIGIT_DBL_SIGNED): Ditto.
(PRI_BDIGIT_PREFIX): Undefine the definition.
(PRI_BDIGIT_DBL_PREFIX): Ditto.

- ext/bigdecimal/bigdecimal.c (RBIGNUM_ZERO_P): Use rb_bigzero_p.
(bigzero_p): Removed.
(is_even): Use rb_big_pack.

Revision 41502 - 06/20/2013 03:42 PM - akr (Akira Tanaka)

- ext/bigdecimal: Workaround fix for bigdecimal test failures caused by [ruby-dev:47413] [Feature #8509]

- ext/bigdecimal/bigdecimal.h (BDIGIT): Make it independent from the definition for bignum.c.
(SIZEOF_BDIGITS): Ditto.
(BDIGIT_DBL): Ditto.
(BDIGIT_DBL_SIGNED): Ditto.
(PRI_BDIGIT_PREFIX): Undefine the definition.
(PRI_BDIGIT_DBL_PREFIX): Ditto.

- ext/bigdecimal/bigdecimal.c (RBIGNUM_ZERO_P): Use rb_bigzero_p.
(bigzero_p): Removed.
(is_even): Use rb_big_pack.

History

#1 - 06/10/2013 11:43 PM - naruse (Yui NARUSE)

- Category set to core

- Status changed from Open to Assigned

- Assignee set to akr (Akira Tanaka)

- Target version set to 2.1.0

I'm ok about introducing this.

Anyway as far as I confirm, gcc 4.1 supports __int128_t and __uint128_t on x64.

#2 - 06/11/2013 12:23 AM - mrkn (Kenta Murata)

I think it should be merged.

I'm trying to change the bigdecimal's precision treatment
so that it's independent of the size of internal type (such as BDIGIT).

Bigdecimal's precision should be the number of decimal figures.
After it's done, these failures are removed.

On Mon, Jun 10, 2013 at 11:43 PM, naruse (Yui NARUSE) naruse@airemix.jp wrote:

> Issue #8509 has been updated by naruse (Yui NARUSE).
>
> Category set to core
> Status changed from Open to Assigned
> Assignee set to akr (Akira Tanaka)
> Target version set to current: 2.1.0
>
> I'm ok about introducing this.

Anyway as far as I confirm, gcc 4.1 supports __int128_t and __uint128_t on x64.

Feature #8509: Use 128 bit integer type in Bignum
https://bugs.ruby-lang.org/issues/8509#change-39842

Author: akr (Akira Tanaka)
Status: Assigned
Priority: Normal
Assignee: akr (Akira Tanaka)
Category: core
Target version: current: 2.1.0

How about Bignum uses 128 bit integer type?

I found that recent gcc (since gcc 4.6) supports 128 bit integer type,
__int128, on some platforms.
http://gcc.gnu.org/gcc-4.6/changes.html

It seems gcc supports it on x86_64 and not on i386.

Currently Ruby implements Bignum on top of 32 bit integer type (BDIGIT)
and 64 bit integer type (BDIGIT_DBL).
(Ruby uses two integer types for multiplication.
BDIGIT_DBL can represent any value of BDIGIT * BDIGIT.)

Historically, Ruby supported platforms without 64 bit integer type.
Ruby used 16 bit integer type (BDIGIT) and 32 bit integer type (BDIGIT_DBL)
on such platform.
However I guess no one use such platforms today.

So with gcc 4.6 or later, we can use 64 bit integer type (BDIGIT) and
128 bit integer type (BDIGIT_DBL).

This may gain performance.

I implemented it. (int128-bignum.patch)

Simple benchmark on Debian GNU/Linux 7.0 (wheezy) x86_64:

trunk% time ./ruby -e 'v = 3*1000; u = 1; 1000.times { u *= v }'
./ruby -e 'v = 31000; u = 1; 1000.times { u *= v }' 1.64s user 0.00s system 99% cpu 1.655 total
128bit% time ./ruby -e 'v = 31000; u = 1; 1000.times { u *= v }'
./ruby -e 'v = 3*1000; u = 1; 1000.times { u *= v }' 1.21s user 0.01s system 99% cpu 1.222 total

I think larger integer type reduces control overhead and compiler will have more opportunity for optimization.

However the patch has API incompatibility.

BDIGIT and BDIGIT_DBL and related definitions are defined in a public headers,
ruby/defines.h.

So third party extensions may be broken with the change.

Note that BDIGIT_DBL is a macro (not typedef name), compiler used for third party extension
don't need to support __int128 unless the extension actually uses BDIGIT_DBL.

If a program try to extract information from a Bignum and assumes BDIGIT is 32 bit integer,
the result may be invalid.
In this situation rb_big_pack/rb_big_unpack or rb_integer_pack/rb_integer_unpack [ruby-core:55408] may help.

However BDIGIT size change itself may cause problems.

One example I patched is about rb_big_pow.
int128-bignum.patch contains following modification for rb_big_pow.

- const long BIGLEN_LIMIT = BITSPERDIG*1024*1024;
- const long BIGLEN_LIMIT = 32*1024*1024;

BIGLEN_LIMIT controls the rb_big_pow generates a Bignum or a Float.
If it is not modified, a test causes memory allocation failure.

Another problem is bigdecimal tests.
bigdecimal tests failed with int128-bignum.patch as follows.

1) Failure:
TestBigDecimal#test_power_of_three [/home/akr/tst1/ruby/test/bigdecimal/test_bigdecimal.rb:1006]:
<(1/81)> expected but was
<#>.

2) Failure:
TestBigDecimal#test_power_with_prec [/home/akr/tst1/ruby/test/bigdecimal/test_bigdecimal.rb:1110]:
<#> expected but was
<#>.

3) Failure:
TestBigDecimal#test_power_without_prec [/home/akr/tst1/ruby/test/bigdecimal/test_bigdecimal.rb:1103]:
<#> expected but was
<#>.

4) Failure:
TestBigDecimal#test_sqrt_bigdecimal [/home/akr/tst1/ruby/test/bigdecimal/test_bigdecimal.rb:796]:
expected but was
<#>.

5) Failure:
TestBigMath#test_atan [/home/akr/tst1/ruby/test/bigdecimal/test_bigmath.rb:60]:
[ruby-dev:41257].
<#> expected but was
<#>.

I guess bigdecimal determines precision depend on sizeof(BDIGIT).
I think it is not a good way to use BDIGIT.

How do you think, mrkn?

Also, we cannot define PRI_BDIGIT_DBL_PREFIX because
there is no printf directive for __int128.

Anyway, is Bignum with __int128 worth to support?
Any opinion?

--
http://bugs.ruby-lang.org/

--
Kenta Murata
OpenPGP FP = 1D69 ADDE 081C 9CC2 2E54 98C1 CEFE 8AFB 6081 B062

□□□□□□□□!!
□Ruby □□□□□□□□ http://www.amazon.co.jp/dp/4798119881/mrkn-22

E-mail: mrkn@mrkn.jp
twitter: http://twitter.com/mrkn/
blog: http://d.hatena.ne.jp/mrkn/

#3 - 06/18/2013 06:46 PM - akr (Akira Tanaka)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r41379.
Akira, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- configure.in: Check __int128.

- include/ruby/defines.h (BDIGIT_DBL): Use uint128_t if it is available.
  (BDIGIT): Use uint64_t if uint128_t is available.
  (SIZEOF_BDIGITS): Defined for above case.
  (BDIGIT_DBL_SIGNED): Ditto.
  (PRI_BDIGIT_PREFIX): Ditto.

- include/ruby/ruby.h (PRI_64_PREFIX): Defined.

- bignum.c (rb_big_pow): Don't use BITSPERDIG for the condition which
  rb_big_pow returns Float or Bignum.

[ruby-dev:47413] [Feature #8509]

#4 - 06/18/2013 06:47 PM - akr (Akira Tanaka)

I committed the patch at r41379.

I hope mrkn will fix the BigDecimal test failures soon.

#5 - 11/24/2013 01:55 PM - akr (Akira Tanaka)

I decided to disable __int128 for Bignum because it is not always faster.

__int128 is still be usable by specifying CPPFLAGS for configure as:
configure CPPFLAGS='-DBDIGIT=uint64_t -DSIZEOF_BDIGITS=8 -DBDIGIT_DBL=uint128_t -DBDIGIT_DBL_SIGNED=int128_t
-DSIZEOF_BDIGIT_DBL=16'

Files

int128-bignum.patch	2.69 KB	06/10/2013	akr (Akira Tanaka)