Ruby - Feature #8796

Use GMP to accelerate Bignum operations

08/17/2013 04:10 AM - akr (Akira Tanaka)

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

Description

How about using GMP to accelerate Bignum operations?

GMP: The GNU Multiple Precision Arithmetic Library
http://gmplib.org/

I wrote a simple patch to use GMP to accelerate Bignum multiplication.

If a user don't want to use GMP, a configure option, --without-gmp,
disables this feature.
Since GMP is licensed as LGPL, some people would need it.
However I think most people can accept LGPL as Ruby 1.8's regex engine.
So, my patch uses GMP by default, if it is available.

It converts bignums from RBignum to mpz_t and back for each
large Bignum multiplication.
RBignum structure itself is not changed and ABI compatible.
(So, this is different from ko1's idea mentioned in Feature #6083)

The conversion cost is O(n).
It is negligible for operations slower than O(n) with large inputs.
Multiplication is a kind of such operation.

I measured the performance as follows.

```
% ./ruby -I.ext/x86_64-linux -r-test-/bignum -e '
methods = %i[big_mul_normal big_mul_karatsuba big_mul_toom3 big_mul_gmp]
m = 1000
n1 = 3**60
100.times {
n1 = n1 * (n1 >> (n1.size*8/15*14))
n2 = n1 + 1
bits = n1.size*8
methods.dup.each {|meth|
t1 = Process.clock_gettime(Process::CLOCK_THREAD_CPUTIME_ID, :nanoseconds)
n1.send(meth, n2) rescue next
(m-1).times { n1.send(meth, n2) }
t2 = Process.clock_gettime(Process::CLOCK_THREAD_CPUTIME_ID, :nanoseconds)
t = (t2 - t1)*1e-9 / m
puts "#{bits},#{t},#{meth.to_s.sub(/big_mul_/, "")}"
methods.delete meth if 1.0/m < t
}
STDOUT.flush
}
'
```

It seems GMP is faster when multiplication arguments are longer than 1000 bits
on my environment.
See bignum-mul-gmp.png for details.

I guess other operations, division and radix conversion, can also be faster using GMP.

Any comments?

History

#1 - 08/17/2013 04:53 AM - normalperson (Eric Wong)

"akr (Akira Tanaka)" akr@fsij.org wrote:

> If a user don't want to use GMP, a configure option, --without-gmp,
> disables this feature.
> Since GMP is licensed as LGPL, some people would need it.
> However I think most people can accept LGPL as Ruby 1.8's regex engine.
> So, my patch uses GMP by default, if it is available.

I'm happy with LGPL :)

> It converts bignums from RBignum to mpz_t and back for each
> large Bignum multiplication.
> RBignum structure itself is not changed and ABI compatible.
> (So, this is different from ko1's idea mentioned in Feature #6083)
>
> The conversion cost is O(n).
> It is negligible for operations slower than O(n) with large inputs.
> Multiplication is a kind of such operation.

Is there more performance improvement without the conversion?

How about push the conversion cost to legacy C API users to make
Bignum faster for pure-Ruby use in a future patch?

I'm mainly curious about "smaller" Bignums for users on 32-bit systems,
but I suspect much of that cost is object allocation.

#2 - 08/17/2013 09:53 AM - akr (Akira Tanaka)

2013/8/17 Eric Wong normalperson@yhbt.net:

> Is there more performance improvement without the conversion?
>
> How about push the conversion cost to legacy C API users to make
> Bignum faster for pure-Ruby use in a future patch?

It is same as ko1's idea.
I don't against it.
Feel free to implement and propose it.

However it has several difficulties.

- It is a big task.
  It need to implement all methods, not just slow methods.
- ABI incompatibility.
  ko1 tackles this in Feature #6083.
- LGPL
  It is no problem for me but I guess some people don't accept it.
  So we need to maintain non-GMP implementation anyway.
  Maintaining two implementations is troublesome.
- We cannot access internal of mpz_t.
  We may be limited to add new feature with optimal performance.
  (mpz_getlimbn and mpz_size may be enough?)
- It cannot embed small bignums.
  So it needs more memory allocation.
  (mpz_array_init may solve this problem?)
  --
  Tanaka Akira

#3 - 08/31/2013 03:57 PM - matz (Yukihiro Matsumoto)

- Assignee set to akr (Akira Tanaka)

This is internal. So go ahead and experiment.

Matz.

#4 - 09/26/2013 10:38 AM - naruse (Yui NARUSE)

- Status changed from Open to Closed

- Target version set to 2.1.0

Introduced on r42743.

Files

bignum-mul-gmp.patch	5.04 KB	08/17/2013	akr (Akira Tanaka)
bignum-mul-gmp.png	22.6 KB	08/17/2013	akr (Akira Tanaka)