Ruby master - Bug #13019

Fix st_hash* functions

12/09/2016 02:02 PM - funny_falcon (Yura Sokolov)

Status:	Closed		
Priority:	Normal		
Assignee:	nobu (Nobuyoshi Nakada)		
Target version:			
ruby -v:		**Backport:**	2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN

Description

Previous implementation had an issues:

- macros murmur1 assumes murmur_step takes rotation value as a second argument
- but murmur_step second argument is "next block"
- this makes st_hash_uint and st_hash_end to not mix high bits of hash value into lower bits
- this leads to pure hash behavior on doubles and mixing hashes using st_hash_uint. It didn't matter when bins amount were prime numbers, but it hurts when bins are powers of two.

Mistake were created cause of attempt to co-exist Murmur1 and Murmur2
in a same code.

Change it to single hash-function implementation.

- block function is in a spirit of Murmur functions, but handles inter-block dependency a bit better (imho).
- final block is read in bit more optimal way on CPU with unaligned word access,
- final block is mixed in simple way,
- finalizer is taken from MurmurHash3 (it makes most of magic :)) (64bit finalizer is taken from http://zimbry.blogspot.ru/2011/09/better-bit-mixing-improving-on.html)

Also remove ST_USE_FNV1: it lacks implementation of many functions,
and looks to be abandoned

Associated revisions

Revision c0ff5f4d - 12/21/2016 06:22 AM - nobu (Nobuyoshi Nakada)

st.c: fix st_hash* functions [Bug #13019]

Previous implementation had an issues:

- macros murmur1 assumes murmur_step takes rotation value as a second argument
- but murmur_step second argument is "next block"
- this makes st_hash_uint and st_hash_end to not mix high bits of hash value into lower bits
- this leads to pure hash behavior on doubles and mixing hashes using st_hash_uint. It didn't matter when bins amount were prime numbers, but it hurts when bins are powers of two.

Mistake were created cause of attempt to co-exist Murmur1 and Murmur2
in a same code.

Change it to single hash-function implementation.

- block function is in a spirit of Murmur functions, but handles inter-block dependency a bit better (imho).
- final block is read in bit more optimal way on CPU with unaligned word access,
- final block is mixed in simple way,
- finalizer is taken from MurmurHash3 (it makes most of magic :)) (64bit finalizer is taken from http://zimbry.blogspot.ru/2011/09/better-bit-mixing-improving-on.html)

Also remove ST_USE_FNV1: it lacks implementation of many functions,
and looks to be abandoned

Author: Sokolov Yura aka funny_falcon funny.falcon@gmail.com

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@57134 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 57134 - 12/21/2016 06:22 AM - nobu (Nobuyoshi Nakada)

st.c: fix st_hash* functions [Bug #13019]

Previous implementation had an issues:

- macros murmur1 assumes murmur_step takes rotation value as a second argument
- but murmur_step second argument is "next block"
- this makes st_hash_uint and st_hash_end to not mix high bits of hash value into lower bits
- this leads to pure hash behavior on doubles and mixing hashes using st_hash_uint. It didn't matter when bins amount were prime numbers, but it hurts when bins are powers of two.

Mistake were created cause of attempt to co-exist Murmur1 and Murmur2
in a same code.

Change it to single hash-function implementation.

- block function is in a spirit of Murmur functions, but handles inter-block dependency a bit better (imho).
- final block is read in bit more optimal way on CPU with unaligned word access,
- final block is mixed in simple way,
- finalizer is taken from MurmurHash3 (it makes most of magic :)) (64bit finalizer is taken from http://zimbry.blogspot.ru/2011/09/better-bit-mixing-improving-on.html)

Also remove ST_USE_FNV1: it lacks implementation of many functions,
and looks to be abandoned

Author: Sokolov Yura aka funny_falcon funny.falcon@gmail.com

Revision 57134 - 12/21/2016 06:22 AM - nobu (Nobuyoshi Nakada)

st.c: fix st_hash* functions [Bug #13019]

Previous implementation had an issues:

- macros murmur1 assumes murmur_step takes rotation value as a second argument
- but murmur_step second argument is "next block"
- this makes st_hash_uint and st_hash_end to not mix high bits of hash value into lower bits
- this leads to pure hash behavior on doubles and mixing hashes using st_hash_uint. It didn't matter when bins amount were prime numbers, but it hurts when bins are powers of two.

Mistake were created cause of attempt to co-exist Murmur1 and Murmur2
in a same code.

Change it to single hash-function implementation.

- block function is in a spirit of Murmur functions, but handles inter-block dependency a bit better (imho).
- final block is read in bit more optimal way on CPU with unaligned word access,
- final block is mixed in simple way,
- finalizer is taken from MurmurHash3 (it makes most of magic :)) (64bit finalizer is taken from http://zimbry.blogspot.ru/2011/09/better-bit-mixing-improving-on.html)

Also remove ST_USE_FNV1: it lacks implementation of many functions,
and looks to be abandoned

Author: Sokolov Yura aka funny_falcon funny.falcon@gmail.com

Revision 57134 - 12/21/2016 06:22 AM - nobu (Nobuyoshi Nakada)

st.c: fix st_hash* functions [Bug #13019]

Previous implementation had an issues:

- macros murmur1 assumes murmur_step takes rotation value as a second argument
- but murmur_step second argument is "next block"
- this makes st_hash_uint and st_hash_end to not mix high bits of hash value into lower bits
- this leads to pure hash behavior on doubles and mixing hashes using st_hash_uint. It didn't matter when bins amount were prime numbers, but it hurts when bins are powers of two.

Mistake were created cause of attempt to co-exist Murmur1 and Murmur2
in a same code.

Change it to single hash-function implementation.

- block function is in a spirit of Murmur functions, but handles inter-block dependency a bit better (imho).
- final block is read in bit more optimal way on CPU with unaligned word access,
- final block is mixed in simple way,

- finalizer is taken from MurmurHash3 (it makes most of magic :)) (64bit finalizer is taken from http://zimbry.blogspot.ru/2011/09/better-bit-mixing-improving-on.html)

Also remove ST_USE_FNV1: it lacks implementation of many functions,
and looks to be abandoned

Author: Sokolov Yura aka funny_falcon funny.falcon@gmail.com

Revision 57134 - 12/21/2016 06:22 AM - nobu (Nobuyoshi Nakada)

st.c: fix st_hash* functions [Bug #13019]

Previous implementation had an issues:

- macros murmur1 assumes murmur_step takes rotation value as a second argument
- but murmur_step second argument is "next block"
- this makes st_hash_uint and st_hash_end to not mix high bits of hash value into lower bits
- this leads to pure hash behavior on doubles and mixing hashes using st_hash_uint. It didn't matter when bins amount were prime numbers, but it hurts when bins are powers of two.

Mistake were created cause of attempt to co-exist Murmur1 and Murmur2
in a same code.

Change it to single hash-function implementation.

- block function is in a spirit of Murmur functions, but handles inter-block dependency a bit better (imho).
- final block is read in bit more optimal way on CPU with unaligned word access,
- final block is mixed in simple way,
- finalizer is taken from MurmurHash3 (it makes most of magic :)) (64bit finalizer is taken from http://zimbry.blogspot.ru/2011/09/better-bit-mixing-improving-on.html)

Also remove ST_USE_FNV1: it lacks implementation of many functions,
and looks to be abandoned

Author: Sokolov Yura aka funny_falcon funny.falcon@gmail.com

History

#1 - 12/09/2016 07:14 PM - vmakarov (Vladimir Makarov)

Yura Sokolov wrote:

> Previous implementation had an issues:
>
> - macros murmur1 assumes murmur_step takes rotation value as a second argument
> - but murmur_step second argument is "next block"
> - this makes st_hash_uint and st_hash_end to not mix high bits of hash value into lower bits
> - this leads to pure hash behavior on doubles and mixing hashes using st_hash_uint. It didn't matter when bins amount were prime numbers, but it hurts when bins are powers of two.

Although performance of MRI hash benchmarks are not improved by this patch, I confirm a bad behaviour of the current variant of murmur hash in MRI. A half year ago, I ran it on SMhasher (https://github.com/aappleby/smhasher) and it failed on a few tests although murmur hashes in SMHasher itself passed the tests. So it is better to fix the bad quality for murmur hash in MRI. Sorry, I did not check your variant in SMHasher. I think it would be nice to do this.

#2 - 12/10/2016 10:18 PM - funny_falcon (Yura Sokolov)

It passes SMHasher (at least, little-endian variant).

#3 - 12/21/2016 05:45 AM - shyouhei (Shyouhei Urabe)

- Assignee set to nobu (Nobuyoshi Nakada)

- Status changed from Open to Assigned

#4 - 12/21/2016 06:22 AM - nobu (Nobuyoshi Nakada)

- Status changed from Assigned to Closed

Applied in changeset r57134.

---

st.c: fix st_hash* functions [Bug #13019]

Previous implementation had an issues:

- macros murmur1 assumes murmur_step takes rotation value as a second argument
- but murmur_step second argument is "next block"
- this makes st_hash_uint and st_hash_end to not mix high bits of hash value into lower bits
- this leads to pure hash behavior on doubles and mixing hashes using st_hash_uint. It didn't matter when bins amount were prime numbers, but it hurts when bins are powers of two.

Mistake were created cause of attempt to co-exist Murmur1 and Murmur2
in a same code.

Change it to single hash-function implementation.

- block function is in a spirit of Murmur functions, but handles inter-block dependency a bit better (imho).
- final block is read in bit more optimal way on CPU with unaligned word access,
- final block is mixed in simple way,
- finalizer is taken from MurmurHash3 (it makes most of magic :)) (64bit finalizer is taken from http://zimbry.blogspot.ru/2011/09/better-bit-mixing-improving-on.html)

Also remove ST_USE_FNV1: it lacks implementation of many functions,
and looks to be abandoned

Author: Sokolov Yura aka funny_falcon funny.falcon@gmail.com

Files

0001-st.c-fix-st_hash-functions.patch	10.9 KB	12/09/2016	funny_falcon (Yura Sokolov)