independent_particle benchmark

October 10-11, 2017 (*see also October 12 update below)*

Working with version from the eigen branch:

```
commit 16be0738707de5b7abb6d9574a20e952996d28f1
Author: James Amundson <amundson@fnal.gov>
Date:   Tue Oct 10 17:31:04 2017 -0500

    loosen boost requirement
```

tev

installed environment modules in ~/opt/modules. Added to bashrc:

```
. ~/opt/modules/init/profile.sh
. ~/work/projects/justspack/share/spack/setup-env.sh
```

In order to work with minigia, I do

```
spack load gcc
spack load cmake
```

Various compilers:

gcc 4.4 (system):

```
|tev>cd build-tev-gcc44
|tev>grep DEFINES CMakeCache.txt
DEFINES:UNINITIALIZED=-ffast-math
|tev>./independent_particle
orig best time = 0.005406
optimized best time = 0.003673
optimized speedup = 1.47182
GSVector::implementation = 1
vectorized best time = 0.00191
vectorized speedup = 1.92304
omp simd best time = 0.003671
omp simd speedup = 1.00054
```

gcc 4.8 (/usr/local):

```
|tev>cd build-tev-gcc48/
|tev>grep DEFINES CMakeCache.txt
DEFINES:UNINITIALIZED=-ffast-math
|tev>prefixadd /usr/local/gcc-4.8.2
|tev>ldpathadd /usr/local/gcc-4.8.2/lib64
|tev>./independent_particle
orig best time = 0.00536215
optimized best time = 0.00358196
optimized speedup = 1.49699
GSVector::implementation = 1
vectorized best time = 0.00191678
vectorized speedup = 1.86874
omp simd best time = 0.00358083
omp simd speedup = 1.00032
```

gcc 5.1 (/usr/local):

```
|tev>cd build-tev-gcc51
|tev>grep DEFINES CMakeCache.txt
DEFINES:UNINITIALIZED=-ffast-math
|tev>ldpathadd /usr/local/gcc-5.1.0/lib64
|tev>prefixadd /usr/local/gcc-5.1.0
|tev>./independent_particle
orig best time = 0.00536193
optimized best time = 0.0038617
optimized speedup = 1.38849
GSVector::implementation = 1
vectorized best time = 0.00208626
vectorized speedup = 1.85101
omp simd best time = 0.00198575
omp simd speedup = 1.94471
```

gcc 7.2 (spack):

```
|tev>cd build-tev-gcc72
|tev>spack load gcc
|tev>grep DEFINES CMakeCache.txt
DEFINES:UNINITIALIZED=-ffast-math
|tev>./independent_particle
orig best time = 0.00462292
optimized best time = 0.00358175
optimized speedup = 1.29069
GSVector::implementation = 1
```

```
vectorized best time = 0.00203214
vectorized speedup = 1.76255
omp simd best time = 0.00180544
omp simd speedup = 1.98386
```

intel 2017:

```
|tev>cd build-intel-universal
|tev>grep DEFINES CMakeCache.txt
DEFINES:UNINITIALIZED=-axMIC-AVX512,AVX,SSE4.2,SSE2
|tev>./independent_particle
orig best time = 0.00546011
optimized best time = 0.00357958
optimized speedup = 1.52535
GSVector::implementation = 1
vectorized best time = 0.00203285
vectorized speedup = 1.76087
omp simd best time = 0.001939
omp simd speedup = 1.8461
```

intel12:

In order to get the universal intel binary to run on intel12, I need to ldpathadd on the two libraries in the libs directory, then spack load gcc.

gcc 7.2:

```
|tev0101>cd build-tev-gcc72/
|tev0101>./independent_particle
orig best time = 0.00467056
optimized best time = 0.00362531
optimized speedup = 1.28832
GSVector::implementation = 1
vectorized best time = 0.00176309
vectorized speedup = 2.05622
omp simd best time = 0.00176298
omp simd speedup = 2.05635
```

intel 2017:

```
|tev0101>cd build-intel-universal
|tev0101>grep DEFINES CMakeCache.txt
```

```
DEFINES:UNINITIALIZED=-axMIC-AVX512,AVX,SSE4.2,SSE2
|tev0101>./independent_particle
orig best time = 0.00529191
optimized best time = 0.0035927
optimized speedup = 1.47296
GSVector::implementation = 1
vectorized best time = 0.00179564
vectorized speedup = 2.00079
omp simd best time = 0.00181186
omp simd speedup = 1.98289
```

knl

get to knl with

```
qsub -q knl -l nodes=1:knl -A mi -I
```

need (for intel)

```
ldpathadd ~/extra/lib
. /opt/intel/compilers_and_libraries/linux/bin/compilervars.sh intel64
. /opt/intel/compilers_and_libraries/linux/mpi/intel64/bin/mpivars.sh intel64
MODULEPATH=/home/amundson/work/projects/justspack/share/spack/modules/linux-
scientificfermi6-x86_64:/home/amundson/opt/modules/modulefiles
spack load gcc
spack load cmake
```

(additional for gcc 7.2)

```
spack load openmpi
```

gcc 7.2:

```
|knl1>cd build-knl
|knl1>grep DEFINES CMakeCache.txt
DEFINES:UNINITIALIZED=-ffast-math -march=knl
|knl1>./independent_particle
orig best time = 0.0126429
optimized best time = 0.0101326
optimized speedup = 1.24775
GSVector::implementation = 1
vectorized best time = 0.00441139
vectorized speedup = 2.29692
omp simd best time = 0.00120199
```

icc 2017:

```
|knl1>cd build-knl-icca/
|knl1>grep DEFINES CMakeCache.txt
DEFINES:UNINITIALIZED=-xMIC-AVX512
|knl1>./independent_particle
orig best time = 0.00854646
optimized best time = 0.00718185
optimized speedup = 1.19001
GSVector::implementation = 1
vectorized best time = 0.0052826
vectorized speedup = 1.35953
omp simd best time = 0.000882793
omp simd speedup = 8.13537
```

panal7

added to bashrc:

```
. /home/amundson/work/projects/justspack/share/spack/setup-env.sh
```

to use intel compiler:

```
. /opt/intel/bin/compilervars.sh intel64
```

gcc 4.8 (system):

```
|panal7>cd build-gcc48
|panal7>grep DEFINES CMakeCache.txt
DEFINES:UNINITIALIZED=-ffast-math
|panal7>./independent_particle
orig best time = 0.00425929
optimized best time = 0.00290606
optimized speedup = 1.46566
GSVector::implementation = 1
vectorized best time = 0.00146869
vectorized speedup = 1.97867
omp simd best time = 0.00290487
omp simd speedup = 1.00041
```

gcc 7.2:

```
|panal7>cd build-gcc72
|panal7>grep DEFINES CMakeCache.txt
DEFINES:UNINITIALIZED=-ffast-math
```

```
|panal7>./independent_particle
orig best time = 0.00358477
optimized best time = 0.0029044
optimized speedup = 1.23425
GSVector::implementation = 1
vectorized best time = 0.00142039
vectorized speedup = 2.0448
omp simd best time = 0.00142964
omp simd speedup = 2.03157
```

intel 2017:

```
|panal7>cd build-intel/
|panal7>grep DEFINES CMakeCache.txt
DEFINES:UNINITIALIZED=-axAVX,SSE4.2,SSE2
|panal7>./independent_particle
orig best time = 0.00424219
optimized best time = 0.00288041
optimized speedup = 1.47277
GSVector::implementation = 1
vectorized best time = 0.00145798
vectorized speedup = 1.97561
omp simd best time = 0.00142179
omp simd speedup = 2.0259
```

zlorfik

gcc 4.8 (system):

```
|zlorfik>./independent_particle
orig best time = 0.00355568
optimized best time = 0.00242013
optimized speedup = 1.46921
GSVector::implementation = 1
vectorized best time = 0.00121264
vectorized speedup = 1.99576
omp simd best time = 0.00243985
omp simd speedup = 0.99192
```

gcc 7.2: (note use of AVX, native architecture)

```
|zlorfik>grep DEFINES CMakeCache.txt
DEFINES:UNINITIALIZED=-march=native -ffast-math -DGSV_AVX
```

```
|zlorfik>./independent_particle
orig best time = 0.00299124
optimized best time = 0.00241769
optimized speedup = 1.23723
GSVector::implementation = 2
vectorized best time = 0.00117436
vectorized speedup = 2.05872
omp simd best time = 0.00117715
omp simd speedup = 2.05384
```

intel 2017:

This binary was copied from panal7, along with several shared libraries. I used

```
prefixadd `pwd`
```

before running independent_particle

```
|zlorfik>cd copied-intel
|zlorfik>./independent_particle
orig best time = 0.0035414
optimized best time = 0.00240271
optimized speedup = 1.47392
GSVector::implementation = 1
vectorized best time = 0.00120261
vectorized speedup = 1.9979
omp simd best time = 0.0011753
omp simd speedup = 2.04434
```

pgi 17.4:

Get pgi compilers on zlorfik by using

```
prefixadd /opt/pgi/linux86-64/17.4
```

vectorclass uses intrinsics not available in pgi, so I had to remove the GSVector implementation. See the pgi branch of minigia.

pgi only implements OpenMP 3, so pragma simd just generated a warning.

```
|zlorfik>cd build-pgi
|zlorfik>./independent_particle
orig best time = 0.00479701
optimized best time = 0.00118916
optimized speedup = 4.03394
omp simd best time = 0.00117477
```

```
omp simd speedup = 1.01225
```

My Mac

Everything (cmake, gcc, non-native clang) is from Homebrew.

gcc-7.2:

```
lmac>cd build-gcc72
lmac>grep DEFINES CMakeCache.txt
DEFINES:UNINITIALIZED=-ffast-math -DGSV_AVX -march=native
lmac>./independent_particle
orig best time = 0.001802
optimized best time = 0.001441
optimized speedup = 1.25052
GSVector::implementation = 2
vectorized best time = 0.000781
vectorized speedup = 1.84507
omp simd best time = 0.000781
omp simd speedup = 1.84507
```

apple clang:

```
lmac>cd build-appleclang
lmac>grep DEFINES CMakeCache.txt
DEFINES:UNINITIALIZED=-ffast-math -DGSV_AVX -march=native
lmac>./independent_particle
orig best time = 0.00180926
optimized best time = 0.0014459
optimized speedup = 1.25131
GSVector::implementation = 2
vectorized best time = 0.000782871
vectorized speedup = 1.84692
```

clang-5.0:

```
lmac>cd build-clang5
lmac>grep DEFINES CMakeCache.txt
DEFINES:UNINITIALIZED=-ffast-math -DGSV_AVX -march=native
lmac>./independent_particle
orig best time = 0.00181236
optimized best time = 0.00144507
optimized speedup = 1.25417
GSVector::implementation = 2
```

```
vectorized best time = 0.000782885
vectorized speedup = 1.84582
omp simd best time = 0.000783298
omp simd speedup = 1.84485
```

October 12, 2017

Now using one later revision from eigen branch:

```
commit 415774e272d18dbd2777ef3fea4cc6038595234c
Author: James Amundson <amundson@fnal.gov>
Date:   Thu Oct 12 10:39:09 2017 -0500

    add tests of omp simd on simpler loops
```

Testing with gcc and intel compiler gives unchanged performance using the simpler loop versions. PGI can not auto-vectorize the simpler versions, however. Still need to do a test with intel on KNL.

Appendix

```
lmac>type prefixadd
prefixadd is a function
prefixadd ()
{
    pathadd $1/bin;
    ldpathadd $1/lib
}
lmac>type ldpathadd
ldpathadd is a function
ldpathadd ()
{
    if [ -z "$1" ]; then
        export LD_LIBRARY_PATH=`pwd`:$LD_LIBRARY_PATH;
    else
        export LD_LIBRARY_PATH=$1:$LD_LIBRARY_PATH;
    fi
}
lmac>type pathadd
pathadd is a function
pathadd ()
{
    if [ -z "$1" ]; then
```

```
        export PATH=`pwd`:$PATH;
    else
        export PATH=$1:$PATH;
    fi
}
```