Vectorization
Where do FLOPS come from?

• A single NewRiver (Haswell) core has theoretical peak performance of 40 billion floating point operations (GFLOPS) per second:
  – 2.5 GHz (billion clock cycles/sec) CPU frequency (×2.5e9)
  – Two 256 bit (4 double-precision FP) vector units (×8)
  – Fused multiply-add (FMA) instructions (×2)

• Note: This is per core, so doesn’t account for 24 cores per node.
for (i=0; i<=MAX; i++)
    c[i] = a[i] + b[i];
for (i=0; i<=MAX; i++)
c[i] = a[i] + b[i];
NewRiver (Haswell CPU) has 4-wide DP vector unit:

```
[jkrometi@nr088 jeffers]$ ./helloflops1_novec
Initializing
Starting Compute
Gflops = 25.600, Secs = 6.583, GFlops per sec = 3.889
```

```
[jkrometi@nr088 jeffers]$ ./helloflops1_vec
Initializing
Starting Compute
Gflops = 25.600, Secs = 1.457, GFlops per sec = 17.576
```
How do I vectorize?

• Mostly done by the compiler (loop unrolling)
  – Optimization level 2 (−O2) or greater
• However, the compiler must know how to break things up:
  – Known number of iterations (not data-dependent)
  – Single entry/exit (e.g., no breaks)
  – (Almost) no branching
  – No big function calls
• Best performance for contiguous memory accesses
Vectorization Reports

• Intel 15 (prints to .optrpt):
  
  ```
  icc -qopt-report=3 -qopt-report-phase=vec -O3 code.c -o a.out
  ```

• Intel 13 (prints to screen):
  
  ```
  icc -vec-report=3 -O3 code.c -o a.out
  ```

• GCC 5.2:
  
  ```
  gcc -ftree-vectorizer-verbose=3 -fopt-info-vec=gcc_vec.txt -O3 code.c -o a.out
  ```
Intel 15.3 Example:

LOOP BEGIN at helloflops1.c(53,3)
  remark #15300: LOOP WAS VECTORIZED
  remark #15449: unmasked aligned unit stride stores: 2
  remark #15467: unmasked aligned streaming stores: 2
  remark #15475: --- begin vector loop cost summary ---
  remark #15476: scalar loop cost: 17
  remark #15477: vector loop cost: 6.500
  remark #15478: estimated potential speedup: 2.610
  remark #15479: lightweight vector operations: 16
  remark #15487: type converts: 6
  remark #15488: --- end vector loop cost summary ---
LOOP END
• A Guide to Vectorization with Intel® C++ Compilers:
  https://software.intel.com/sites/default/files/m/4/8/8/2/a/31848-
  CompilerAutovectorizationGuide.pdf

• Auto-vectorization in GCC:
  vectorization.html