

performance/compiler\_optionen/intel\_llvm/example\_vec\_report\_st

## Example: Intel compiler optimization report for benchmark stream

- stream source code snippet

```
/* ---
Tuned vector scale:  b[] = scalar * c[]

In:  STREAM_ARRAY_SIZE_thread, scalar, c[]
Out: b[]
--- */
void static inline tuned_STREAM_Scale(const STREAM_TYPE scalar) {
    #pragma omp parallel default(none) shared(scalar, STREAM_ARRAY_SIZE_thread)
    {
        #ifdef __INTEL_COMPILER
            // Instructs the compiler to use non-temporal (that is, streaming) stores
            #pragma vector nontemporal
        #endif
        #pragma omp simd aligned(b, c : alignment_bytes)
        for (long int j = 0; j < STREAM_ARRAY_SIZE_thread; j++) {
            b[j] = scalar * c[j]; // Line: 349
        }
    }
}

/* ---
Tuned vector add:  c[] = a[] + b[]

In:  STREAM_ARRAY_SIZE_thread, a[], b[]
Out: c[]
--- */
void static inline tuned_STREAM_Add() {
    #pragma omp parallel default(none) shared(STREAM_ARRAY_SIZE_thread)
```

```

{
    #ifdef __INTEL_COMPILER
        // Instructs the compiler to use non-temporal (that is, streaming) stores
        #pragma vector nontemporal
    #endif
    #pragma omp simd aligned(a, b, c : alignment_bytes)
    for (long int j = 0; j < STREAM_ARRAY_SIZE_thread; j++) {
        c[j] = a[j] + b[j]; // Line: 369
    }
}
}

```

- Prepare environment

```

module purge
module add compiler/intel/2022

```

- Compile benchmark with optimization report enabled

```

icx -std=c11 -Ofast -xHost -ipo -qopenmp \
    -qopt-report=max \
    stream.OpenMP.c

```

- Output

```

LOOP BEGIN at stream.OpenMP.c (347, 9)
    remark #15569: Compiler has chosen to target XMM/YMM vector. Try using -mprefer-vec
    remark #15300: LOOP WAS VECTORIZED
    remark #15305: vectorization support: vector length 4
    remark #15475: --- begin vector loop cost summary ---
    remark #15482: vectorized math library calls: 0
    remark #15484: vector function calls: 0
    remark #15485: serialized function calls: 0
    remark #15488: --- end vector loop cost summary ---
    remark #15447: --- begin vector loop memory reference summary ---
    remark #15450: unmasked unaligned unit stride loads: 1
    remark #15451: unmasked unaligned unit stride stores: 1
    remark #15456: masked unaligned unit stride loads: 0
    remark #15457: masked unaligned unit stride stores: 0
    remark #15458: masked indexed (or gather) loads: 0
    remark #15459: masked indexed (or scatter) stores: 0
    remark #15462: unmasked indexed (or gather) loads: 0
    remark #15463: unmasked indexed (or scatter) stores: 0
    remark #15554: Unmasked VLS-optimized loads (each part of the group counted separat
    remark #15555: Masked VLS-optimized loads (each part of the group counted separatel
    remark #15556: Unmasked VLS-optimized stores (each part of the group counted separat
    remark #15557: Masked VLS-optimized stores (each part of the group counted separat
    remark #15474: --- end vector loop memory reference summary ---
LOOP END

```

```

LOOP BEGIN at stream.OpenMP.c (347, 9)
<Remainder loop for vectorization>
    remark #15441: remainder loop was not vectorized:
LOOP END
...

...
Global optimization report for : main.extracted.110

LOOP BEGIN at stream.OpenMP.c (367, 9)
    remark #15569: Compiler has chosen to target XMM/YMM vector. Try using -mprefer-vec
    remark #15300: LOOP WAS VECTORIZED
    remark #15305: vectorization support: vector length 4
    remark #15475: --- begin vector loop cost summary ---
    remark #15482: vectorized math library calls: 0
    remark #15484: vector function calls: 0
    remark #15485: serialized function calls: 0
    remark #15488: --- end vector loop cost summary ---
    remark #15447: --- begin vector loop memory reference summary ---
    remark #15450: unmasked unaligned unit stride loads: 2
    remark #15451: unmasked unaligned unit stride stores: 1
    remark #15456: masked unaligned unit stride loads: 0
    remark #15457: masked unaligned unit stride stores: 0
    remark #15458: masked indexed (or gather) loads: 0
    remark #15459: masked indexed (or scatter) stores: 0
    remark #15462: unmasked indexed (or gather) loads: 0
    remark #15463: unmasked indexed (or scatter) stores: 0
    remark #15554: Unmasked VLS-optimized loads (each part of the group counted separat
    remark #15555: Masked VLS-optimized loads (each part of the group counted separatel
    remark #15556: Unmasked VLS-optimized stores (each part of the group counted separa
    remark #15557: Masked VLS-optimized stores (each part of the group counted separat
    remark #15474: --- end vector loop memory reference summary ---
LOOP END

LOOP BEGIN at stream.OpenMP.c (367, 9)
<Remainder loop for vectorization>
    remark #15441: remainder loop was not vectorized:
LOOP END

    - Report on successful vectorization
    - Report on vector length
    - Report on loads and stores

```