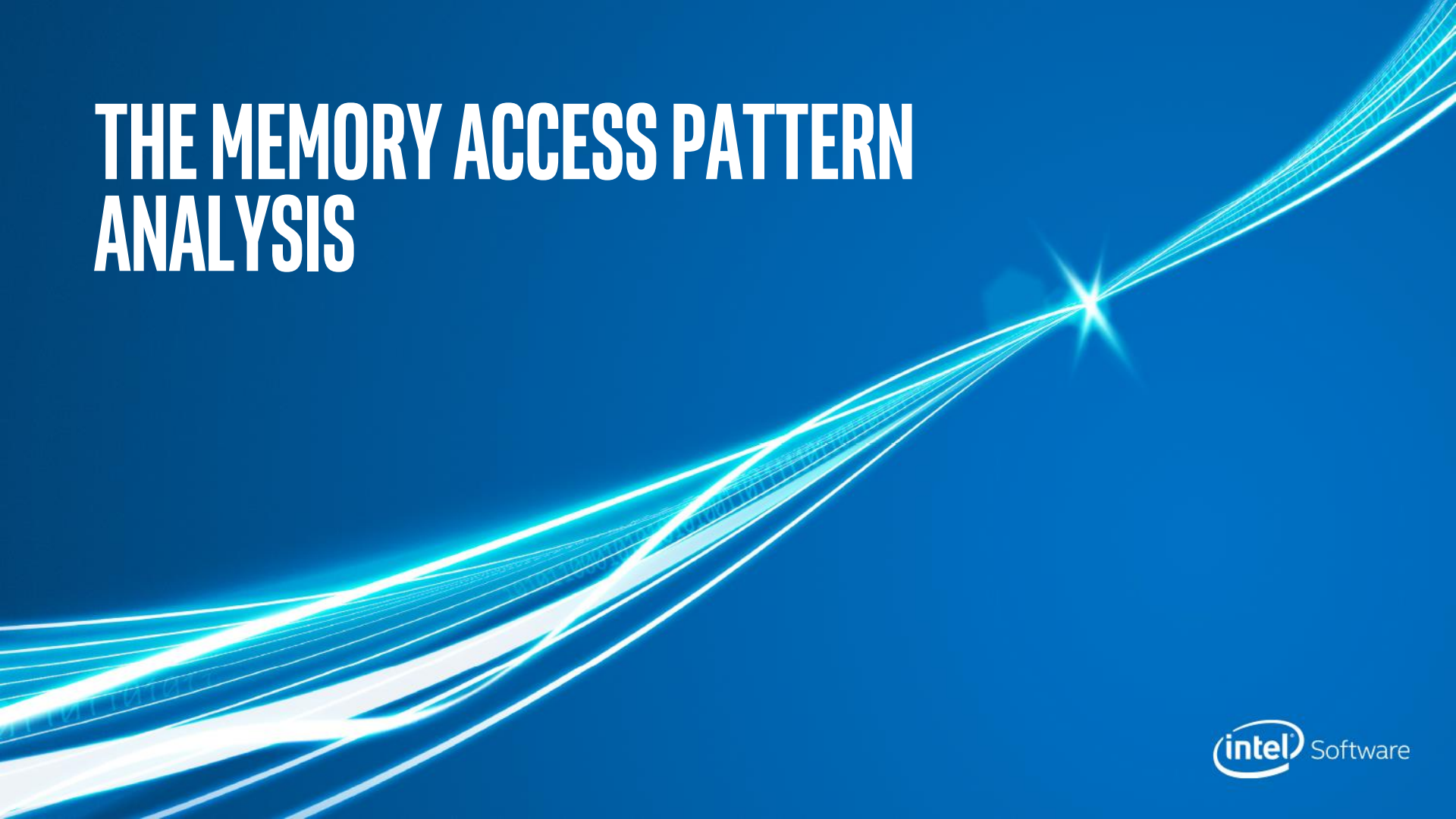
USING INTEL® ADVISOR WITH THE ROOFLINE MODEL TO BOOST YOUR APPLICATIONS

Profiling and improving your vectorization

THE MEMORY ACCESS PATTERN ANALYSIS



MEMORY ACCESS PATTERN ANALYSIS

How should I access data ?

- Unit stride access are faster

```
for (i=0; i<N; i++)
     A[i] = B[i]*d
```

- Constant stride are more complex

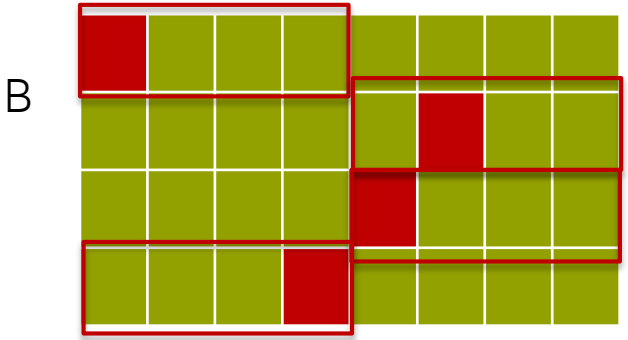
```
for (i=0; i<N; i+=2)
     A[i] = B[i]*d
```

- Non predictable access are usually bad

```
for (i=0; i<N; i++)
     A[i] = B[C[i]]*d
```

For B, 1 cache line load computes 4 DP

For B, 2 cache line loads compute 4 DP with reconstructions

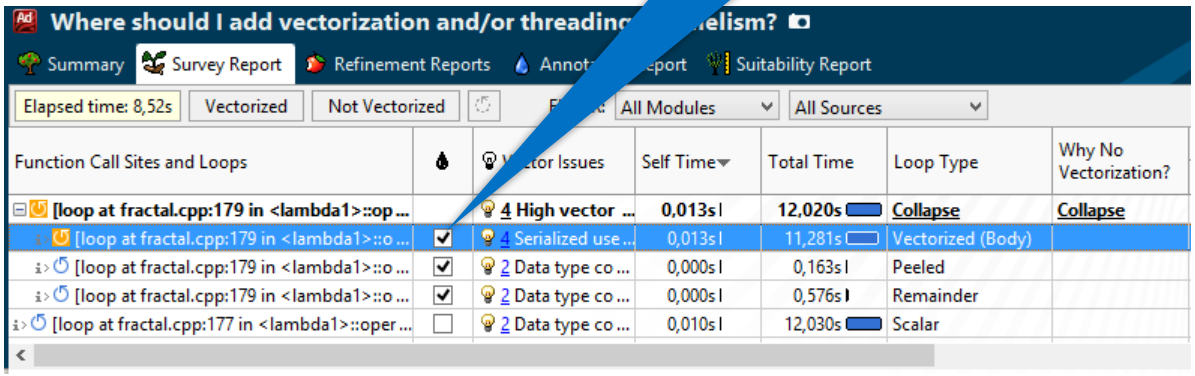


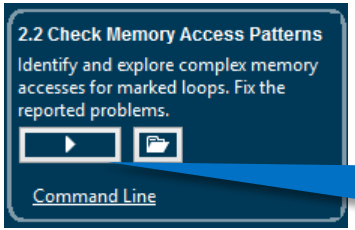
For B, 4 cache line loads compute 4 DP with reconstructions, prefetching might not work

HOW TO RUN IT ?

Automatic or manual modes

Select loops of interest



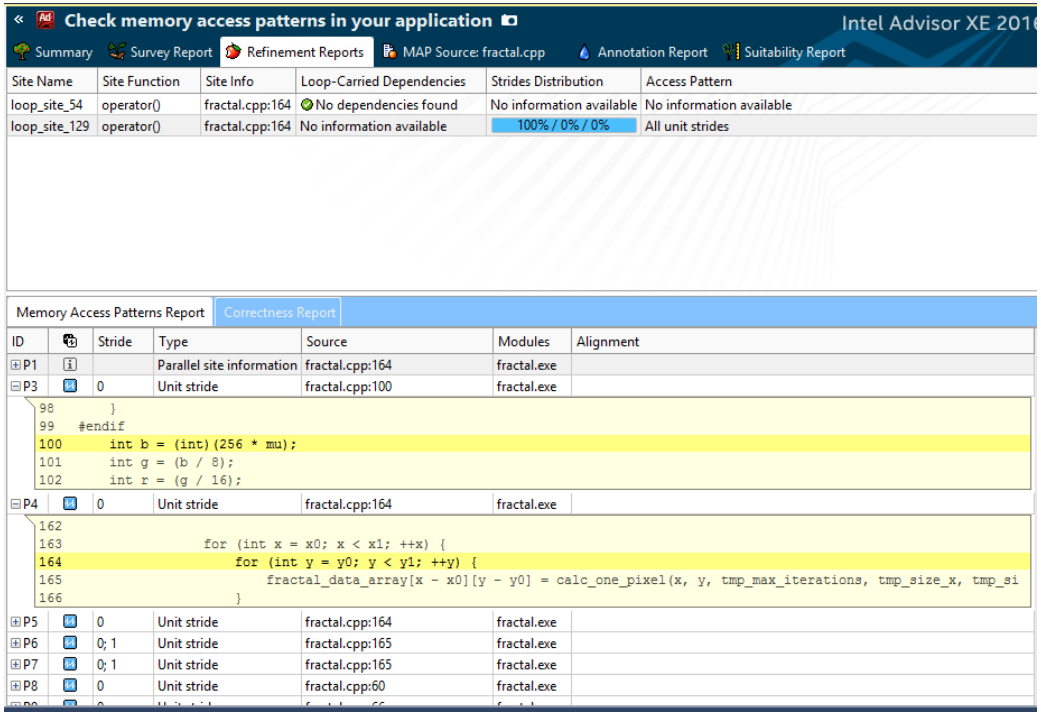


Run Memory Access Patterns analysis, just to check how memory is used in the loop and the called function

QUICKLY FIND LOOP WITH NON-OPTIMAL STRIDE

Identify the problem first

- Quickly identify loops that are good, bad or mixed
- Find unaligned data
- `pragma omp declare simd uniform(…)`



IMPROVING DATA ACCESS

What can you do to improve vectorization ?

Constant stide

- Verify loop order
- Move from AoS to SoA
  - Rewrite your code completely
  - Use Intel® SIMD Data Layout Templates (Intel® SDLT)

Unpredictable memory accesses

- Very difficult case
- Algorithm changes might be required
- Reorganize the datalayout


intel
Software