1 Auto Vectorization

1.1 Characteristics of SSE, AVX(2) and AVX-512

SSE (Streaming SIMD Extensions):

Introduced with x86, supports 128-bit vector registers, enabling SIMD operations on 4 single-precision floating-point or 2 double-precision values in parallel. Limited in width and functionality compared to later extensions.

AVX/AVX2 (Advanced Vector Extensions):

Extends SIMD width to 256 bits, doubling parallelism compared to SSE. AVX introduced floating-point operations, while AVX2 added integer operations and support for gather instructions.

AVX-512:

Widens SIMD to 512 bits, allowing operations on 16 single-precision floats or 8 double-precision values simultaneously. It includes more registers, mask registers for conditional operations, and broader instruction sets but increases power consumption and requires hardware support.

1.2 Impact of Memory Aliasing on Performance

Memory aliasing occurs when two or more pointers reference overlapping memory regions, making it difficult for the compiler or CPU to optimize memory accesses. This uncertainty can prevent reordering or parallelizing memory operations, resulting in stalls and reduced performance due to additional dependency checks and cache conflicts.

1.3 Advantages of Unit Stride Memory Access

Unit stride (stride-1) memory access sequentially accesses adjacent memory locations, maximizing cache utilization and enabling efficient prefetching. Larger strides, like stride-8, result in less efficient cache usage, increased memory latency, and potential cache line thrashing due to non-contiguous data access patterns.

1.4 When to Prefer Structure of Arrays

Structure of Arrays (SoA) is preferred when working with SIMD/vectorized operations or when the workload requires processing individual fields of a dataset independently. SoA provides better memory alignment and avoids unnecessary memory fetches, leading to improved cache efficiency and parallelism compared to Array of Structures (AoS).