HPC-I Lab1 Automatic Vectorization

• Student ID: 111032042

• Name: 盛爾葳

SIMD Instruction Set	Time	Screenshot (including compilation logs)
SSE	4330 ms	**Id81811004(princessionicil)2 **/workspaces/MPC-I_MBO (main) \$ g++ -o labilabi.cpp -03 -funrol l-loops -fout-frame-pointer -finilme-functions -miss -fopt-info-wc-all labil.cpp -03 -funrol l-loops -fout-frame-pointer -finilme-functions -miss -fopt-info-wc-all labil.cpp -103 -finilme-functions -miss -fort-info-wc-all labil.cpp -103 -finilme-functions -miss -fort-info-wc-all labil.cpp -103 -finilme-function
AVX2	1001 ms	<pre>% -18372456880:circustationsicil22 → /workspaces/INC-1,1800 (main) \$ g+* oo labb labbl.cop -03 -funno 11-loops -fount-frame-pointer -finline-functions -mav2 -fopt-info-vec-all labbl.cop13120: missed: couldn't vectorize looped access gattern. labbl.cop13120: missed: couldn't vectorize looped access gattern. labbl.cop13120: missed: couldn't vectorize looped access gattern. labbl.cop13120: missed: couldn't vectorize looped labbl.cop13120: missed: couldn't vectorize in function. labbl.cop13120: missed: statement clobbers memory: 3 = rand (); labbl.cop13120: missed: statement clobbers memory: 3 = rand (); labbl.cop13120: missed: statement clobbers memory: 4 = rand (); labbl.cop13120: missed: statement clobbers memory: 4 = rand (); labbl.cop13120: missed: statement clobbers memory: 4 = rand (); labbl.cop13120: missed: statement clobbers memory: 4 = rand (); labbl.cop13120: missed: statement clobbers memory: 4 = rand (); labbl.cop13120: missed: statement clobbers memory: 4 = rand (); labbl.cop13120: missed: statement clobbers memory: 4 = rand (); labbl.cop13120: missed: statement clobbers memory: 4 = rand (); labbl.cop13120: missed: statement clobbers memory: 4 = rand (); labbl.cop13120: missed: statement clobbers memory: 4 = rand (); labbl.cop13120: missed: statement clobbers memory: 50 = statement clobbers memory: 50 =</pre>
AVX512 (if your computer supports AVX512)	Not supported	* optimizes winnishi2 → /workspaces/HPC-I_NBO (main) \$ gs + oo labl labl.cpp -03 *funroll-loops of ont-frame-pointer -finline-functions -max52f -fopt-infor-wec-all labl.cpp -03 *funroll-loops of ont-frame-pointer -finline-functions -max52f -fopt-infor-wec-all labl.cpp -103 *funroll-loops of ont-frame-pointer -finline-functions -finli

Compare SSE, AVX2 and AVX512

1. SSE Compilation

- Vectorized using 16-byte vectors
- ∘ Execution took 4330 ms

 Some loops could not be vectorized due to complex access patterns or functions like rand() that clobber memory.

2. AVX2 Compilation:

- Vectorized using 32-byte vectors
- ∘ Execution took 1001 ms
- Significantly faster than SSE, likely due to more data being processed per vector operation.

3. AVX-512 Compilation:

- Vectorized using 64-byte vectors
- Encountered "Illegal instruction (core dumped)" upon execution, indicating the CPU does not support AVX-512 instructions, or there was an issue with how these instructions were utilized.

Flags for optimization

 -03: Enables all the optimizations that the compiler offers, excluding those that increase compilation time substantially.

-funroll-loops

- Tells the compiler to unroll loops where it deems beneficial for performance
- Reduces the overhead of loop control but increases the size of the binary.

• -fomit-frame-pointer

- Omits the frame pointer for a slight performance benefit in some cases.
- Make debugging harder but can also free up a register for general use in certain architectures.
- -finline-functions: Encourages the compiler to inline functions, which can reduce the overhead of function calls but might increase the size of the binary.
- -fopt-info-vec-all: Provides detailed vectorization optimization reports from the compiler, showing which loops have been vectorized.