SIMD 命令 でループを高速に その2

SIMD とは(復習)

- 1 命令で複数のデータに演算を適用できる
- 専用のレジスタを使うことが多い
  - 512-bit * 32 みたいな
- たとえば...の高速化に向いている
  - データの総和を求める
  - ベクトル，行列同士の演算

ところで

- コンパイラには自動最適化機能がある
- 今回はそれを掘り下げていく

-O0と-O3の比較

- -O0 (最適化レベル0)

```
gcc -O0 -mavx2 simd.c && ./a.out
vecadd: 2.940000s
vecadd_avx2: 0.770000s
```

- -O3 (最適化レベル3)

```
gcc -O3 -mavx2 simd.c && ./a.out
vecadd: 0.120000s
vecadd_avx2: 0.160000s
```

- あれ？

自動ベクトル化なし

```
vecadd:
.LFB0:
        pushq   %rbp
        movq    %rsp, %rbp
        movq    %rdi, -24(%rbp)
        movq    %rsi, -32(%rbp)
        movq    %rdx, -40(%rbp)
        movl    $0, -4(%rbp)
        jmp     .L2
.L3:
        movl    -4(%rbp), %eax
        cltq
        leaq    0(,%rax,4), %rdx
        movq    -32(%rbp), %rax
        addq    %rdx, %rax
        vmovss  (%rax), %xmm1
        movl    -4(%rbp), %eax
        cltq
        leaq    0(,%rax,4), %rdx
        movq    -40(%rbp), %rax
        addq    %rdx, %rax
        vmovss  (%rax), %xmm0
        movl    -4(%rbp), %eax
        cltq
        leaq    0(,%rax,4), %rdx
        movq    -24(%rbp), %rax
        addq    %rdx, %rax
        vaddss  %xmm0, %xmm1, %xmm0
        vmovss  %xmm0, (%rax)
        addl    $1, -4(%rbp)
.L2:
        cmpl    $1023, -4(%rbp)
        jle     .L3
        nop
        popq    %rbp
        ret
```

自動ベクトル化あり

- めっちゃ長いので省略(`-O3 -mavx2` とかつけて自分で見てみて)
- `-O0` の場合と比較して
  - `vaddss` のようなAVXのスカラ命令だけでなく，`vaddps` のようなベクトル命令も活用されている

コンパイラによる自動ベクトル化

- -O3 とかつけるとコンパイラが決まった形のループなどに自動でSIMD命令を適用してくれる
  - `vaddps` など，ベクトル命令が使われていることがわかる
- 一般的にコンパイラによる自動最適化は優秀なので，組み込み関数(intrinsics)による手動最適化は開発の初期段階から行うべきではない
  - 間違いを埋め込みやすく，バグの温床
  - > The real problem is that programmers have spent far too much time worrying about efficiency in the wrong places and at the wrong times; premature optimization is the root of all evil (or at least most of it) in programming. - Donald Knuth

じゃあ先週のはなんだったのか？

- コンパイラによる自動最適化はときとしてうまく働かない
- 賢い人間が最大限intrinsicsを用いてチューニングしてあげたほうが最高のパフォーマンスを引き出せる場合がある
  - 線型代数関連のライブラリ実装を見てみると，SIMDや浮動小数点命令を使ってゴリゴリにチューニングされてることが多い
    - OpenBLASとか