

Векторные вычисления в C++

Костеров Максим 21 апреля

Векторные функции

SIMD

SIMD

Single Instruction Multiple Data

Одна команда Несколько данных

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} \circ \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} a_1 \circ b_1 \\ a_2 \circ b_2 \\ a_3 \circ b_3 \\ a_4 \circ b_4 \end{pmatrix}$$

SIMD

SIMD

Single Instruction Multiple Data

Одна команда Несколько данных

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} \circ \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} a_1 \circ b_1 \\ a_2 \circ b_2 \\ a_3 \circ b_3 \\ a_4 \circ b_4 \end{pmatrix}$$

SIMD

SIMD

Single Instruction Multiple Data

Одна команда Несколько данных

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} \circ \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} a_1 \circ b_1 \\ a_2 \circ b_2 \\ a_3 \circ b_3 \\ a_4 \circ b_4 \end{pmatrix}$$

SIMD

SIMD

Single Instruction Multiple Data

Одна команда Несколько данных

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} \circ \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} a_1 \circ b_1 \\ a_2 \circ b_2 \\ a_3 \circ b_3 \\ a_4 \circ b_4 \end{pmatrix}$$

Четыре операции \circ за время одной операции \circ

Иструкции AVX

Иструкции AVX

Заголовок `immintrin.h`

Документация

Иструкции AVX

Заголовок `immintrin.h`

Документация

256-битные типы данных

Иструкции AVX

Заголовок `immintrin.h`

Документация

256-битные типы данных

`__m256`

8 float

`__m256d`

4 double

Иструкции AVX

Заголовок `immintrin.h`

Документация

256-битные типы данных

`__m256`

8 float

`__m256d`

4 double

`__m256i`

32 char

16 short

8 int

4 long long

Инструкции AVX

Инструкции, которые мы будем использовать

Инструкции AVX

Инструкции, которые мы будем использовать

1. Загрузка данных

- (a) `__m256d _mm256_setzero_pd()`
`dst[0]=0; dst[1]=0; dst[2]=0; dst[3]=0;`
- (b) `__m256d _mm256_set1_pd(double v)`
`dst[0]=v; dst[1]=v; dst[2]=v; dst[3]=v;`
- (c) `__m256d _mm256_set_pd(double a, double b,
double c, double d)`
`dst[0]=d; dst[1]=c; dst[2]=b; dst[3]=a;`
- (d) `__m256d _mm256_loadu_pd(const double* a)`
`dst[i]=a[i];`

Иструкции AVX

2. Выгрузка данных

(a) `__m256d _mm256_storeu_pd(double *dst, __m256d a)`
`dst[i]=a[i];`

Иструкции AVX

2. Выгрузка данных

(a) `__m256d _mm256_storeu_pd(double *dst, __m256d a)`
`dst[i]=a[i];`

3. Арифметические операции

(a) `__m256d _mm256_add_pd(__m256d a, __m256d b)`
`dst[i]=a[i]+b[i];`

(b) `__m256d _mm256_mul_pd(__m256d a, __m256d b)`
`dst[i]=a[i]*b[i];`

Примеры