

# ИССЛЕДОВАНИЕ АВТОМАТИЧЕСКОЙ ВЕКТОРИЗАЦИИ НА ПРИМЕРЕ КОМПИЛЯТОРА MSVC 14.28

Пикиреня П.И., Русакович А.В.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Перцев Д.Ю. – канд. техн. наук

Продемонстрированы некоторые проблемы, возникающие при работе с автоматической векторизации циклов компилятором MSVC 14.28.

Составная часть компилятора, отвечающая за автоматическое распараллеливание и векторизацию, обеспечивают существенное повышение производительности исполнения цикла в коде. Векторизация – вид распараллеливания программы, при котором однопоточные приложения, выполняющие одну операцию на одном элементом в каждый момент времени, модифицируются для выполнения одной операции сразу над некоторым множеством данных.

Примерами векторных инструкций являются SSE2, AVX и AVX2 в процессорах Intel и AMD [1], NEON в процессорах ARM. Простейший пример, демонстрирующий работу векторизации показан на рисунке 1. Слева показано простое сложение двух векторов на языке программирования C. Справа – векторная версия данного алгоритма, при этом число итераций массива уменьшено в 4 раза, т.к. инструкции movups и addps работают сразу с четырьмя элементами массива.

<pre>float* a, b, c; // инициализация a, b, c пропущена for (unsigned i = 0; i &lt; 1000; i++) {     c[i] = a[i] + b[i]; }</pre>		<pre>mov ecx, 255 mov offset, 0 loop1: movups xmm1, [MEM1 + offset] movups xmm2, [MEM2 + offset] addps xmm1, xmm0 movups [MEM3 + offset], xmm1 add offset, 16 loop loop1</pre>
--	--	--

Рисунок 1 – Пример работы алгоритма векторизации

При замене ограничителя числа итераций (в примере выше – 1000) на константу size, поведение компилятора изменяется и зависит от следующих условий:

- 1) данное значение определено как константа или изменяется
- 2) ее размер.

Чем меньше значение size, тем меньше вероятность успешной автоматической векторизации (например, при значении size = 4 несмотря на то, что вручную векторизовать можно, авторам так и не удалось заставить компилятор векторизовать вручную). При динамическом изменении размера компилятор генерирует две версии кода и определяет, можно ли исполнить векторную версию кода во время исполнения кода. В зависимости от результата выполняется векторная версия (в приоритете) либо традиционная скалярная.

Также неожиданный результат был получен при исполнении кода на рисунке. 2:

```
for (int l = 0; l < size; ++l) {
    c[l] += a[q] * b[l];
}
```

Рисунок 2 – Пример кода с ошибкой векторизации

Несмотря на то, что все условия векторизации выполнены, была получена ошибка векторизации с кодом 1203 (не смежный доступ к массиву [2]). Объяснения, почему так происходит, найдено не было.

Представленный материал будет полезен студентам, изучающим основы векторизации.

#### Список использованных источников:

1. Автоматическая параллелизация и автоматическая векторизация [Электронный ресурс] / Microsoft. – Режим доступа: <https://docs.microsoft.com/ru-ru/cpp/parallel/auto-parallelization-and-auto-vectorization?view=msvc-160#auto-vectorizer3>. Дата доступа: 27.03.2021.

2. Сообщения векторизатора и параллелизатора [Электронный ресурс] / Microsoft. – Режим доступа: <https://docs.microsoft.com/ru-ru/cpp/error-messages/tool-errors/vectorizer-and-parallelizer-messages?view=msvc-160>. Дата доступа: 27.03.2021.