Курсовая работа

Оптимизация ClickHouse под современный набор инструкций CPU

Ковальков Дмитрий Андреевич, БПМИ 175

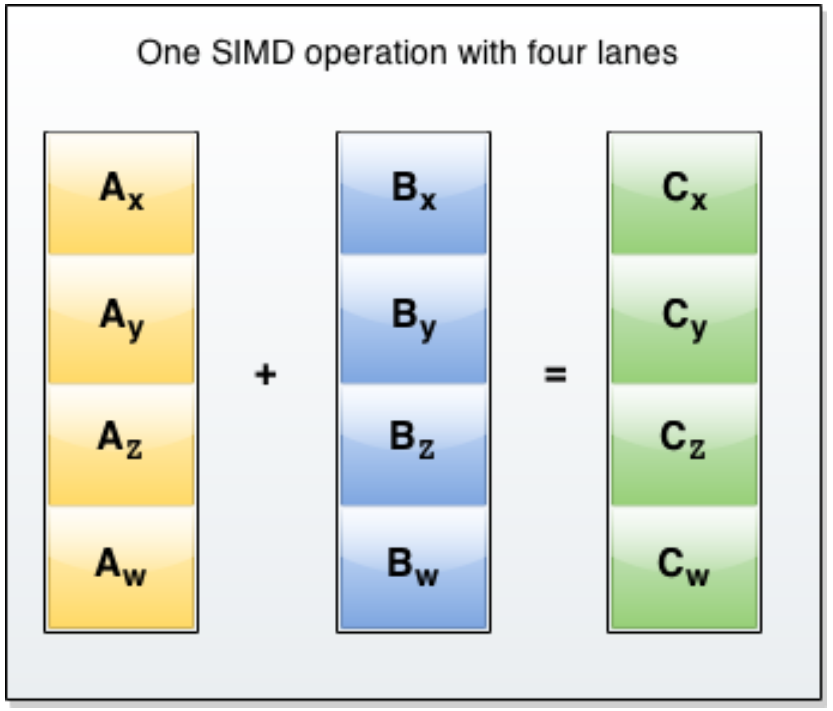
Научный руководитель: Руководитель группы разработки ClickHouse,

Миловидов Алексей Николаевич

Современные наборы инструкций

Большое количество SIMD инструкций, позволяющих обрабатывать множество данных за одну операцию

- SSE4.2
- AVX
- AVX2
- AVX512
- NEON



Актуальность задачи

Особенности ClickHouse:

- Скорость работы
- Колоночное хранение данных
- Используются векторные инструкции до SSE4.2

ClickHouse

Сложности задачи

- Наборов инструкций много – под каждый нужна своя реализация
- Старые и дешевые процессоров не поддерживает все современные наборы инструкций
- Платформо-зависимый код сложен в разработке и поддержке, легко ошибиться
- Включение платформо-зависимого кода специфично для каждого компилятора

Цели и задачи

Необходимо разработать:

- Механизмы включения платформо-зависимого кода в проект
- Механизм генерации кода под разные платформы
- Механизм выбора между реализациями во время исполнения
- Реализации некоторых функций с использованием современных наборов инструкций

Реализация

Вставка и генерация платформо-зависимого кода с помощью макросов.

Все специфичные для каждой платформы объекты и функции находятся в специальных пространствах имен.

В случае, если код не имеет смысла для какой-либо архитектуры, он автоматически удаляется

```
DECLARE_MULTITARGET_CODE(
int myFunc() {
    return 0;
}
) // DECLARE_MULTITARGET_CODE

DECLARE_AVX2_SPECIFIC_CODE(
int myFunc2() {
    return 1;
}
) //DECLARE_AVX2_SPECIFIC_CODE

DECLARE_AVX512F_SPECIFIC_CODE(
int myFunc2() {
    return 2;
}
) // DECLARE_AVX512F_SPECIFIC_CODE
```

Реализация

Класс ImplementationSelector для выбора подходящей реализации.

Проверка текущей платформы происходит автоматически

Выбор из доступных реализаций происходит с помощью метода Байесовских многоруких бандитов.

```
class MyFunc
{
public:
    MyFunc()
    {
        selector.registerImplementation<TargetArch::Default,
            TargetSpecific::Default::MyFunc>();
        selector.registerImplementation<TargetArch::AVX2,
            TargetSpecific::AVX2::MyFunc>();
    }

    void execute(Block & block)
    {
        selector.selectAndExecute(block);
    }

private:
    ImplementationSelector<IFunction> selector;
};
```

Экспериментальные результаты

SELECT count() FROM numbers(100000000) WHERE NOT ignore(intHash32(number))

	Время, default	Время, AVX2	Относительное ускорение
intHash32, gcc, remote	0.1837 с	0.1357 с	26.2%
intHash32, gcc, local	0.3885 с	0.2739 с	29.5%
intHash32, clang, local	0.4610 с	0.3028 с	34.3%
intHash64, gcc, remote	0.1557 с	0.1285 с	17.5%
intHash64, gcc, local	0.3802 с	0.3095 с	18.6%
intHash64, clang, local	0.3225 с	0.2903 с	10.0%
rand, gcc, remote	0.0504 с	0.0390 с	22.7%
rand, gcc, local	0.1235 с	0.1110 с	10.1%
rand, clang, local	0.1205 с	0.1073 с	10.9%

Результаты

Были реализованы механизмы, упрощающие разработку платформо-зависимого кода в ClickHouse.

С помощью данных механизмов была произведена оптимизация ряда функций.

Спасибо за внимание