Боль и страдания в мире оптимизации для RISC-V

Оптимизация xxHash для RISC-V с использованием различных реализаций RVV

Николай Пономарев

Математико-механический факультет СПбГУ

20 сентября 2024 г.

Обо мне

Надо ???

С чего всё начиналось (весна 2023)

- Большой интерес лаборатории к векторному расширению RISC-V
- И наивное ожидание SBC с поддержкой RVV 1.0
- В качестве подопытного — библиотека xxHash
  - Алгоритм хеширования, поддерживающий векторизацию
  - Готовые реализации для SSE2, AVX, NEON, SVE
  - Использование intrinsic функций для оптимизации

Что умеет RISC-V

- Векторное расширение RISC-V ≡ RVV
- В природе встречается две версии:
  - RVV 0.7.1 — в ядрах Xuantie C906 (Sipeed Lichee RV, MangoPi MQ) и C910 (Beagle V, Sipeed Lichee Pi 4A)
  - RVV 1.0 — в ядрах Xuantie C908 и C920, SpacemiT X60 (Banana Pi BPI-F3)
- Программные инструменты не готовы поддерживать RVV 0.7.1

$\Longrightarrow$ для работы с RVV 0.7.1 требуются инструменты напрямую от вендора

Первые эксперименты — QEMU (весна 2023)

- Зачем использовать плату, если есть эмулятор?
- Поддержка только RVV 1.0
- Векторные операции target архитектуры исполняются на скалярных регистрах host устройства

$\Longrightarrow$ QEMU — инструмент тестирования **корректности**, не быстродействия

Первые эксперименты — Lichee RV (весна 2023)

- Sipeed Lichee RV с ядрами Xuantie C906
- RVV 0.7.1 с поддержкой элементов размером 32 бита и меньше
- Проблема 1: чем компилировать?
- Решение 1: будем использовать форк GCC 10 от Xuantie
- Проблема 2: xxHash использует элементы по 64 бита
- Решение 2: используем 32-битные элементы
- Проблема 3: внутри алгоритма используется сложение, нужно помнить про перенос
- Решение 3: будем таскать за собой перенос, но это потребует масок

$\Longrightarrow$ получим слишком много лишних действий $\Longrightarrow$ получить нормальную скорость **невозможно**

Полноценный RVV 0.7.1 (осень 2023)

- Sipeed LicheePi 4A с Xuantie C910 и полноценным RVV 0.7.1
- Код стал проще и более похож на уже существующий
- К этому времени в апстриме компиляторов поменялись названия intrinsic функций
- Пришлось использовать макросы для тестирования в QEMU
- Однако ускорения не получилось
- Возможная проблема — дороговизна инструкции перестановки элементов вектора

$\Longrightarrow$ дальнейшие эксперименты были отложены в дальний ящик

RVV 1.0! (лето 2024)

- BPI-F3 с SpacemiT X60 и RVV 1.0
- Апрстримовые компиляторы!
- И наконец-то ускорение!

$\Longrightarrow$ потребовалось почти 3 года с момента принятия RVV 1.0, чтобы суметь провести оптимизации для него

Бенчмарки

	Скалярная версия, Мб/с	Векторная версия, Мб/с	Ускорение, раз
Урезанный RVV 0.7.1	169.3	116.0	0.69
Полноценный RVV 0.7.1	645.3	472.6	0.73
RVV 1.0	516.3	2036.0	3.94

???

Выводы на тему развития экосистемы???