

Оптимизация xxHash для RISC-V с использованием различных реализаций RVV

Длинная дорога к ускорению маленькой библиотеки для RISC-V

Николай Пономарев

Математико-механический факультет СПбГУ

20 сентября 2024 г.

С чего всё начиналось (весна 2023)

- Большой интерес лаборатории к векторному расширению RISC-V

С чего всё начиналось (весна 2023)

- Большой интерес лаборатории к векторному расширению RISC-V
- И наивное ожидание SBC с поддержкой RVV 1.0

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

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

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

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 \equiv 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

Что умеет RISC-V

- Векторное расширение RISC-V \equiv 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

⇒ для работы с RVV 0.7.1 требуются инструменты напрямую от вендора

Первые эксперименты — QEMU (весна 2023)

- Зачем использовать плату, если есть эмулятор?

Первые эксперименты — QEMU (весна 2023)

- Зачем использовать плату, если есть эмулятор?
- Поддержка только RVV 1.0

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

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

⇒ QEMU — инструмент тестирования **корректности**, не быстродействия

Первые эксперименты — Lichee RV (весна 2023)

- Sipeed Lichee RV с ядрами Xuantie C906

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

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?

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

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

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

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

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

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

⇒ получим слишком много лишних действий ⇒ получить нормальную скорость **невозможно**

Полноценный RVV 0.7.1 (осень 2023)

- Sipeed LicheePi 4A с Xuantie C910 и полноценным RVV 0.7.1

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

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

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

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

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

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

⇒ дальнейшие эксперименты были отложены в дальний ящик

RVV 1.0! (лето 2024)

- BPI-F3 с SpacemiT X60 и RVV 1.0

RVV 1.0! (лето 2024)

- BPI-F3 с SpacemiT X60 и RVV 1.0
- Апстримовые компиляторы!

RVV 1.0! (лето 2024)

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

RVV 1.0! (лето 2024)

- BPI-F3 с SrasemiT X60 и RVV 1.0
- Апстримовые компиляторы!
- И наконец-то ускорение!

⇒ 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

- Появление бета версий расширений в доступном «железе» ведёт к усложнению поддержки кода

Выводы

- Появление бета версий расширений в доступном «железе» ведёт к усложнению поддержки кода
- Между возможностью скомпилировать код и измерить ускорение может пройти достаточно большое количество времени

Выводы

- Появление бета версий расширений в доступном «железе» ведёт к усложнению поддержки кода
- Между возможностью скомпилировать код и измерить ускорение может пройти достаточно большое количество времени
- Нельзя утверждать, что на другом ядре получится добиться ускорения

Выводы

- Появление бета версий расширений в доступном «железе» ведёт к усложнению поддержки кода
- Между возможностью скомпилировать код и измерить ускорение может пройти достаточно большое количество времени
- Нельзя утверждать, что на другом ядре получится добиться ускорения

Спасибо!