

Санкт-Петербургский государственный университет

Кафедра информационно-аналитических систем

Группа 21.Б10-мм

Оптимизация библиотеки XXHASH для архитектуры RISC-V

Пономарев Николай Алексеевич

Отчёт по учебной практике
в форме «Эксперимент»

Научный руководитель:
ст. преподаватель кафедры ИАС К. К. Смирнов

Санкт-Петербург
2023

Оглавление

| | |
|---|----|
| Введение | 3 |
| 1. Постановка задачи (обязателен к Новому году) | 4 |
| 2. Обзор (обязателен к новому году) | 5 |
| 3. Реализация | 6 |
| 4. Эксперимент (желательно к Новому году) | 7 |
| 4.1. Условия эксперимента | 7 |
| 4.2. Исследовательские вопросы | 7 |
| 4.3. Метрики | 8 |
| 4.4. Результаты | 8 |
| 4.5. Обсуждение результатов | 9 |
| Заключение | 10 |
| Список литературы | 11 |

Введение

xxHASH — современная библиотека для хеширования, целью которой является генерация хеша со скоростью, сравнимой со скоростью оперативной памяти [1]. Высокую скорость работы, в частности, для хешей ХХН3 и ХХН128, обеспечивает реализация алгоритмов хеширования с помощью векторных расширений процессора.

1. Постановка задачи (обязателен к Новому году)

Дословно «Целью работы является... Для её выполнения были поставлены следующие задачи:»

1. реализовать это;
2. спроектировать это;
3. протестить на том-то;
4. ~~изучить язык ОСАМ~~ писать тут не надо, так как тут должны быть задачи, выполнение которых можно проверить/оценить прочитав текст или выслушав доклад;
 - ~~произведен обзор предметной области~~ не нужно писать по той же причине. *Исключение: вы опубликовали обзорную статью и готовы её предъявить как доказательство проведенного обзора.*

2. Обзор (обязателен к новому году)

Обзор должен быть. Здесь нужно писать, что индустрия и наука уже сделали по вашей теме. Он нужен, чтобы Вы случайно не изобрели какой-нибудь велосипед.

По-английски называется `related works` или `previous works`.

Если Ваша работа является развитием предыдущей и плохо понимаема без неё, то обзор должен идти в начале. Если же Вы решаете некоторую задачу новым интересным способом, то если поставить обзор в начале, то читатель может устать, пока доберется до вашего решения. В этом случае уместней поставить обзор после описания Вашего подхода к проблеме.

В обзоре необходимо ссылаться на работы других людей. В данном шаблоне задумано, что литература будет указываться в файле `vr.kr.bib`. В нём указываются пункты литературы в формате BIVTEX, а затем на них можно ссылаться с помощью `\cite{...}`. Та литература, на которую Вы сошлётесь, попадет в список литературы в конце документа. Если не сошлётесь — не попадёт. Спецификацию в формате BIVTEX почти никогда (для второго курса — никогда), не нужно придумывать руками. Правильно: находить в интернете описание цитируемой статьи¹, копировать цитату с помощью кнопки “Export Citation” и вставлять в BIVTEX файл. Если так не делать, но оформление литературы будет обрастать багами. Например, BIVTEX по особенному обрабатывает точки, запятые и `and` в списке авторов, что позволяет ему самому понимать, сколько авторов у статьи, и что там фамилия, что — имя, а что — отчество.

В обзоре и в остальном тексте вы наверняка будете использовать названия продуктов или языков программирования. Для них рекомендуется (в файле `preamble2.tex`) задать специальные команды, чтобы писать сложные названия правильно и одинаково по всему документу. Написать с ошибкой название любимого языка программирования научного руководителя — идеальный вариант его выбесить.

¹ Например, <https://dl.acm.org/doi/10.1145/3408995> (дата доступа: 17 декабря 2022 г.).

3. Реализация

Библиотека содержит в себе четыре алгоритма хеширования: XXH32, XXH64, XXH3, XXH128. Первые два из них не поддаются оптимизации с помощью векторных операций, поэтому интерес представляют только последние два. Внутри они используют функции `XXH3_accumulate` и `XXH3_scrambleAcc`. Именно эти функции могут использовать векторные возможности процессора: в библиотеке уже имеется поддержка SSE2, AVX512, NEON и других.

Уже существующие реализации оперируют компонентами вектора размером в 64 бита, в терминологии RISC-V это называется SEW². А длина вектора разнится от 128 бит до 512 бит, в терминологии RISC-V — VL³.

По спецификации векторного расширения RISC-V, минимальный VL равен 128 битам, так же должна присутствовать поддержка SEW равного 64 битам. К сожалению, на момент написания данной работы в продаже можно было найти лишь устройства на чипе ALLWINNER D1, в котором отсутствует поддержка 64-битных элементов вектора, поэтому было решено использовать SEW в 32 бита из-за чего потребовалась некоторое количество ухищрений.

В качестве эталонной реализации был выбран для набора команд SSE2, т.к. целевой процессор имеет такой же VL, как и SSE2.

Одной из первых проблем стала операция умножения. В алгоритме требуется перемножить 2 вектора, используя только младшие 32 бита каждого элемента, в результате чего получаются 64-битные числа. В силу ограничений, использовалась следующая последовательность команд: сначала вычислялись

²Selected Element Width

³Vector Length

4. Эксперимент (желательно к Новому году)

Как мы проверяем, что всё удачно получилось. К Новому году для промежуточного отчета желательно хотя бы описать как он будет проводиться и на чем.

4.1. Условия эксперимента

Железо (если актуально); версии ОС, компиляторов и параметры командной строки; почему мы выбрали именно эти тесты; входные данные, на которых проверяем наш подход, и почему мы выбрали именно их.

4.2. Исследовательские вопросы

По-английски называется *research questions*, в тексте можно ссылаться на них как RQ1, RQ2, и т. д. Необходимо сформулировать, чего мы хотели бы добиться работой (2 пункта будет хорошо):

- Хотим алгоритм, который лучше вот таких-то остальных.
- Если в подходе можно включать/выключать составляющие, то насколько существенно каждая составляющая влияет на улучшения.
- Если у нас строится приближение каких-то штук, то на сколько точными будут эти приближения.
- и т.п.

Иногда в работах это называют гипотезами, которые потом проверяют. Далее в тексте можно ссылаться на *research questions* как RQ, это общепринятое сокращение.

4.3. Метрики

Как мы сравниваем, что результаты двух подходов лучше или хуже:

- Производительность.
- Строчки кода.
- Как часто алгоритм «угадывает» правильную классификацию входа.

Иногда метрики вырожденные (да/нет), это не очень хорошо, но если в области исследований так принято, то ладно.

4.4. Результаты

Результаты понятно что такое. Тут всякие таблицы и графики, как в таблице 1. Обратите внимание, как цифры выровнены по правому краю, названия по центру, а разделители \times и \pm друг под другом.

Скорее всего Ваши измерения будут удовлетворять нормальному распределению, в идеале это надо проверять с помощью критерия Колмогорова и т.п. Если критерий этого не подтверждает, то у Вас что-то сильно не так с измерениями, надо проверять кэши процессора, отключать Интернет во время измерений, подкручивать среду исполнения (англ. runtime), чтобы сборка мусора не вмешивалась и т.п. Если критерий удовлетворён, то необходимо либо указать мат. ожидание и доверительный/предсказывающий интервал, либо написать, что все измерения проводились с погрешностью, например, в 5%. Замечание: если у вас получится улучшение производительности в пределах погрешности, то это обязательно вызовет вопросы.

В этом разделе надо также коснуться Research Questions.

4.4.1. RQ1

Пояснения

Таблица 1: Производительность какого-то алгоритма при различных разрешениях картинок (меньше — лучше), в мс., CI=0.95. За пример таблички кидаем чепчики в честь Я. Кириленко

| Resolution | TENG | LAPM | VOLL4 |
|------------|---------------|---------------|---------------|
| 1920×1080 | 406.23 ± 0.94 | 134.06 ± 0.35 | 207.45 ± 0.42 |
| 1024×768 | 145.00 ± 0.47 | 39.68 ± 0.10 | 52.79 ± 0.10 |
| 464×848 | 70.57 ± 0.20 | 19.86 ± 0.01 | 32.75 ± 0.04 |
| 640×480 | 51.10 ± 0.20 | 14.70 ± 0.10 | 24.00 ± 0.04 |
| 160×120 | 2.40 ± 0.02 | 0.67 ± 0.01 | 0.92 ± 0.01 |

4.4.2. RQ2

Пояснения

4.5. Обсуждение результатов

Чуть более неформальное обсуждение, то, что сделано. Например, почему метод работает лучше остальных? Или, что делать со случаями, когда метод классифицирует вход некорректно.

Заключение

Обязательно для промежуточного, полугодового, годового и любых других отчётов. Кратко, что было сделано.

Для практик/ВКР. Также важно сделать список результатов, который будет один к одному соответствовать задачам из раздела 1.

- Результат к задаче 1
- Результат к задаче 2
- и т.д.

Для промежуточных отчетов сюда важно записать какие задачи уже были сделаны за осенний семестр, а какие только планируется сделать.

Также сюда можно написать планы развития работы в будущем, или, если их много, выделить под это отдельную предпоследнюю главу.

Список литературы

- [1] GitHub - Cyan4973/xxHash: Extremely fast non-cryptographic hash algorithm — github.com. — <https://github.com/Cyan4973/xxHash>. — [Accessed 10-May-2023].