Лабораторная работа “хеши”

Ельчинов Е. С. (Б05-932)

20 декабря 2020 г.

1 Хеш-таблицы

1.1 Реализация

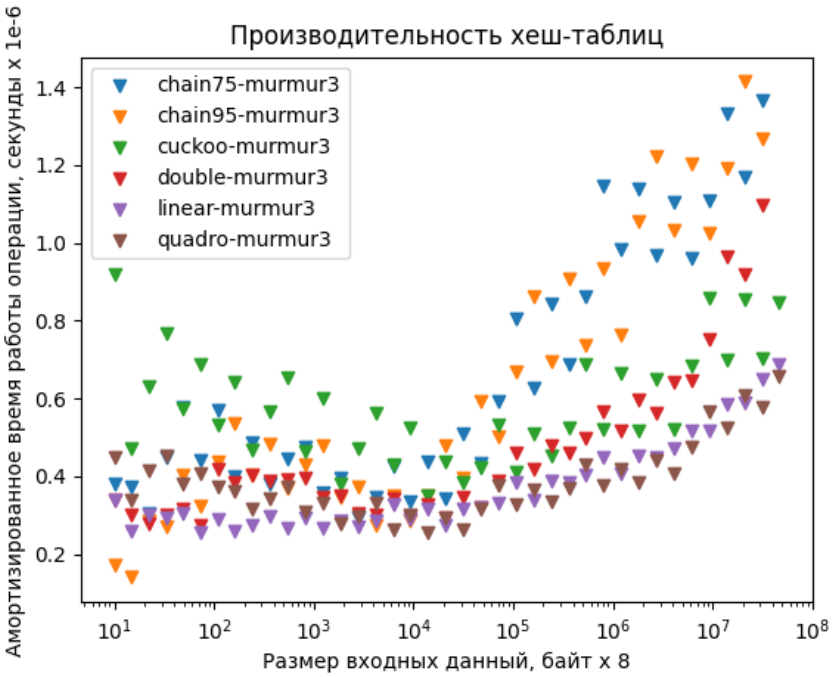
Реализована и протестирована иерархия классов хеш-словарей, параметризуемых хранимыми данными и типом хеш-функции. Были реализованы все требуемые в задании хеш-функции для случая строк и чисел. В бенчмарках используются следующие хеш-функции для случая чисел:

- std::hash
- md5 из библиотеки openssl
- sha256 из библиотеки openssl
- murmur3
- tabulation hashing
- polynomial hashing

1.2 Бенчмарки

Описанные в задании бенчмарки проведены для хеширования чисел типа uint64_t. Для сравнения производительности различных хеш-таблиц был взят murmur3 хеш, как наиболее просто вычисляющийся и гарантированно эффективный. В случае необходимости второго хеша использовался std::hash.

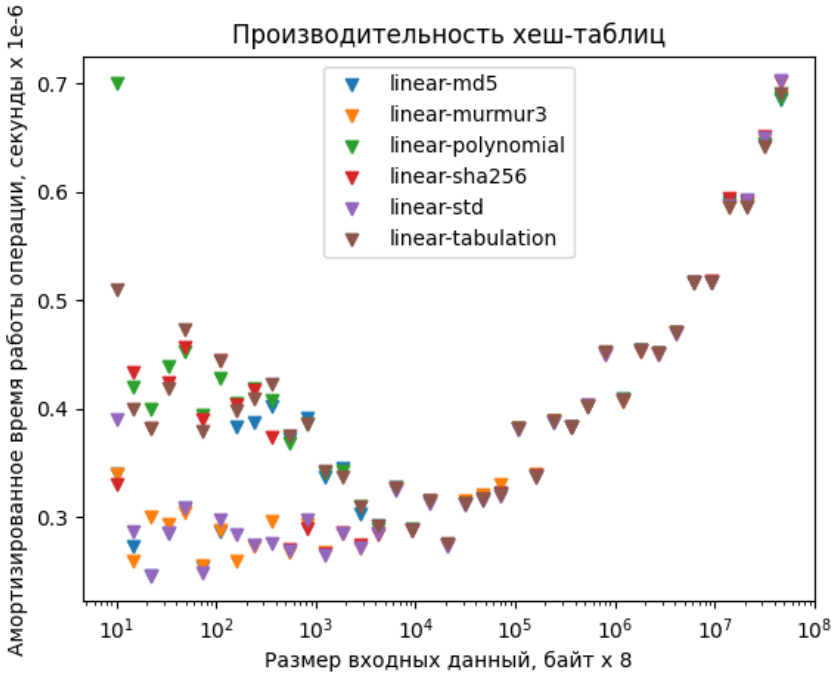
Результаты бенчмарка (время измеряется в микросекундах):



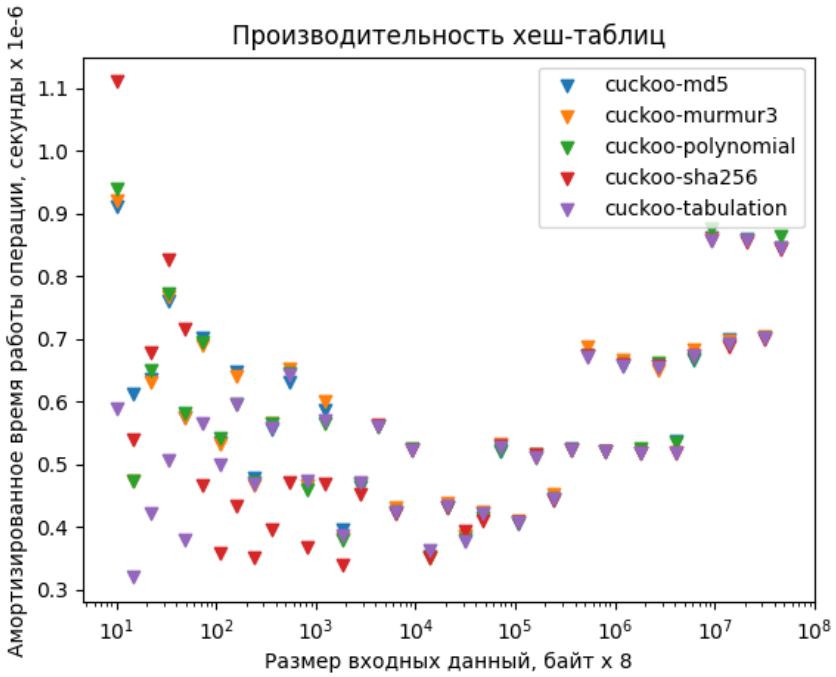
Видно, что таблицы с открытым хешированием и линейным и квадратичным пробированием имеют сравнимую эффективность. Чуть менее эффективна реализация cuckoo-пробирования, а хеш-таблицы, основанные на методе цепочек для разрешения коллизий имеют худшее время работы при достаточно большом объеме хранимых данных.

Бенчмарки для различных хеш-функций проводились на реализации с линейным и cuckoo пробированием.

Линейное пробирование (время измеряется в микросекундах):



Cuckoo пробирование (время измеряется в микросекундах):



Характерные скачки на графиках могут быть объяснены необходимостью затратной операции перехеширования при расширении таблицы.

1.3 Выводы

По результатам бенчмарков, на данным порядков размера кеша 2-го и 3-го уровней лучший результат показывают алгоритмы разрешения коллизий с открытой адресацией и линейным либо квадратичным пробированием и алгоритмы хеширования murmur3 и std::hash в используемой реализации стандартной библиотеки. Меньшая эффективность cuckoo хеширования может объясняться затратностью операции вставки и необходимостью использовать две хеш-функции. Алгоритмы SHA256 и MD5 имеют немного меньшую эффективность из-за избыточной длины возвращаемого хеша и большей сложности, обеспечивающей криптографическую стойкость.