

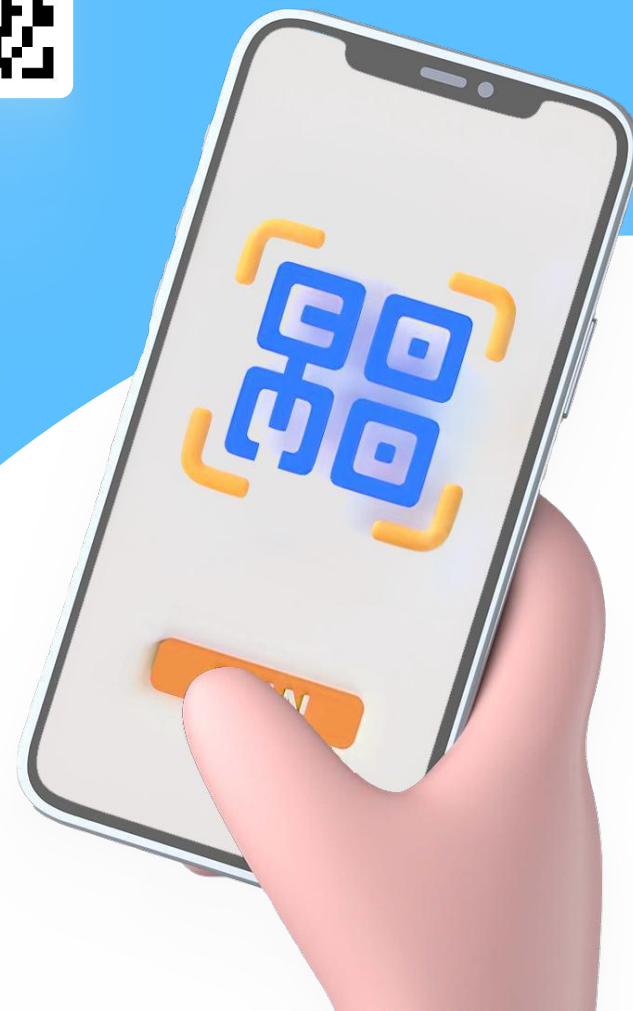


КУПЛЕНО НА
SKLADCHIK.ORG

Все самые свежие
курсы по лучшей цене!



Отсканируйте
QR-код телефоном



Swiss table

Проблемы текущей реализации хэш-таблицы:

- Медленный поиск: сканирование цепочек переполнения означало прыжки в памяти (привет, промахи кэша!)
- Раздувание памяти: переполненные бакеты съедали дополнительное пространство
- Проблемы с изменением размера: увеличение карты означало переиндексирование всего

В хэш-таблице с открытой адресацией все элементы хранятся в одном массиве. Мы будем называть каждое местоположение в массиве слотом.

Слот, к которому принадлежит ключ, в первую очередь определяется хэш-функцией `hash(key)`. Определяющей особенностью хэш-таблиц с открытой адресацией является то, что они разрешают коллизии, сохраняя ключ в другом месте резервного массива.

Таким образом, если слот уже заполнен (коллизия), то пробирование используется для рассмотрения других слотов, пока не будет найден пустой слот.

Swiss table также является формой хэш-таблицы с открытой адресацией. У нас по-прежнему есть один резервный массив для хранения, но мы разобьем массив на логические группы по 8 слотов в каждой

У нас по-прежнему есть один резервный массив для хранения, но мы разобьем массив на логические группы по 8 слотов в каждой

	Group 0								Group 1							
Slot	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Key	56	32	21						78							

	64-bit control word 0								64-bit control word 1							
Slot	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
h2	23	89	50						47							

Кроме того, каждая группа имеет 64-битное контрольное слово для метаданных. Каждый из 8 байтов в контрольном слове соответствует одному из слотов в группе. Значение каждого байта обозначает, является ли этот слот пустым, удаленным или используется

Статус:

0: Empty.

1: Occupied.

0x80 (binary 10000000): Tombstone (deleted).

Вставка работает следующим образом:

- Вычислить хэш и разбить хэш на две части: верхние 57 бит (называемые $h1$) и нижние 7 бит (называемые $h2$).
- Верхние биты ($h1$) используются для выбора первой группы для рассмотрения
- Внутри группы все слоты в равной степени могут содержать ключ. Сначала мы должны определить, содержит ли какой-либо слот этот ключ, в этом случае это обновление, а не новая вставка.
- Если ни один слот не содержит ключ, то мы ищем пустой слот для размещения этого ключа.
- Если ни один слот не пуст, то мы продолжаем последовательность пробирования, выполняя поиск в следующей группе

	Group 0								Group 1							
Slot	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Key	56	32	21						78							

	64-bit control word 0								64-bit control word 1							
Slot	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
h2	23	89	50						47							

Поиск следует тому же базовому процессу. Если мы находим пустой слот на шаге 4, то мы знаем, что вставка использовала бы этот слот, и можем остановить поиск.

Шаг 3 — это то, где происходит магия Swiss Table. Нам нужно проверить, содержит ли какой-либо слот в группе нужный ключ.

Наивно, мы могли бы просто выполнить линейное сканирование и сравнить все 8 ключей. Однако контрольное слово позволяет нам сделать это более эффективно.

Каждый байт содержит нижние 7 бит хэша (h_2) для этого слота. Если мы определим, какие байты контрольного слова содержат h_2 , который мы ищем, у нас будет набор кандидатов на совпадения.

Другими словами, мы хотим выполнить побайтовое сравнение на равенство в управляющем слове.
Например, если мы ищем ключ 32, где $h_2 = 89$, то
нужная нам операция выглядит так.

Test word	89	89	89	89	89	89	89	89
Comparison	==	==	==	==	==	==	==	==
Control word	23	89	50	-	-	-	-	-
Result	0	1	0	0	0	0	0	0

Это операция, поддерживаемая оборудованием SIMD, где одна инструкция выполняет параллельные операции над независимыми значениями внутри большего значения (вектора).

Эта операция очень мощная, поскольку мы фактически выполнили 8 шагов последовательности зондирования одновременно, параллельно.

Результатом является набор слотов-кандидатов. Слоты, где h_2 не совпадает, не имеют соответствующего ключа, поэтому их можно пропустить.

Слоты, где h_2 совпадает, являются потенциальными совпадениями, но мы все равно должны проверить весь ключ, так как существует вероятность коллизий

Переиндексация

Когда хэш-таблица достигает максимального коэффициента загрузки, ей необходимо увеличить массив. Обычно это означает, что следующая вставка удваивает размер массива и копирует все записи в новый массив.

Представьте себе вставку в хэш-таблицу с 1 ГБ записей. Большинство вставок выполняются очень быстро, но одна вставка, которая должна увеличить хэш-таблицу с 1 ГБ до 2 ГБ, должна будет скопировать 1 ГБ записей, что займет много времени.

Go решает эту проблему с помощью еще одного уровня косвенности, разделяя каждую хэш-таблицу на несколько Swiss Tables.

Вместо одной Swiss Table, реализующей всю хэш-таблицу, хэш-таблица состоит из одной или нескольких независимых Swiss Tables.

Отдельная Swiss Table хранит максимум 1024 записи. Переменное количество верхних битов в хэше используется для выбора таблицы, к которой принадлежит ключ. Это форма расширяемого хэширования, где количество используемых битов увеличивается по мере необходимости.

used	
	groups

- power-of-two **groups**
- 8 **slots** per group
- up to 1024 slots in a **table**

- **8 slots** per group

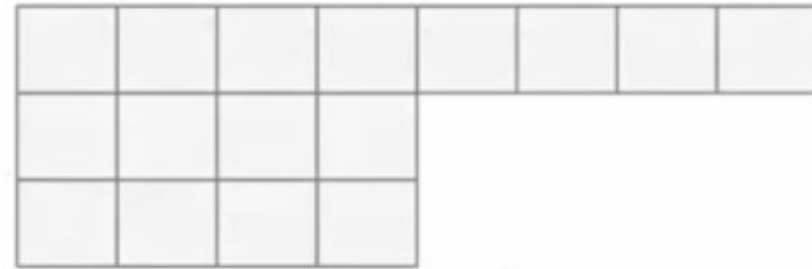
- up to 1024 slots in a **table**

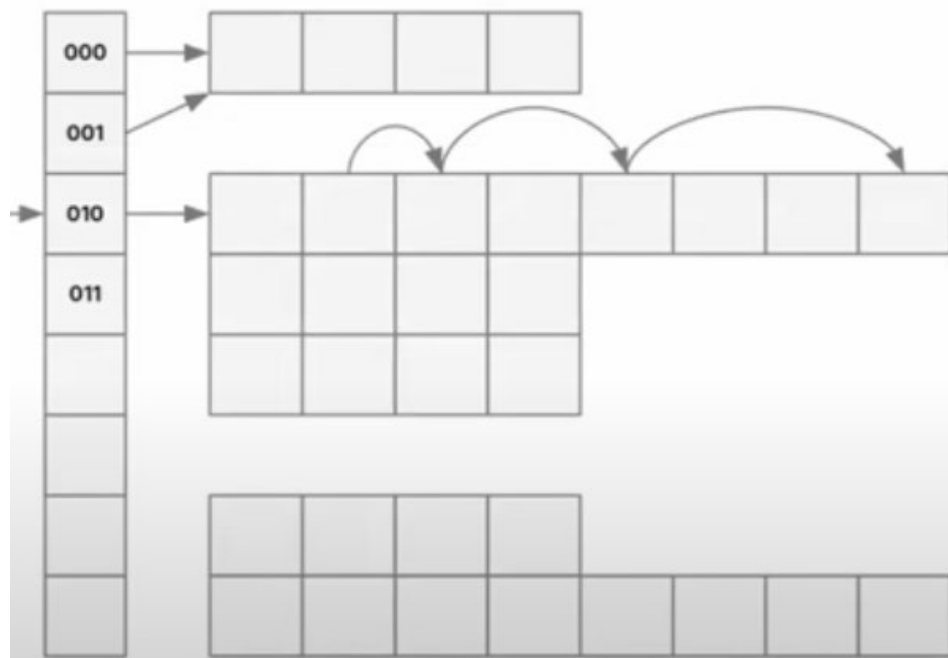
[illegible]

group 1

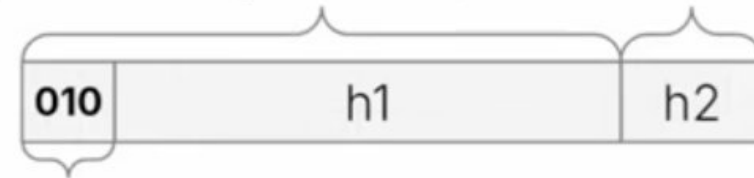
A directory of tables

used	seed
dirPtr	dirLen
globalDepth	





probe seq (57 bits) control word (7 bits)

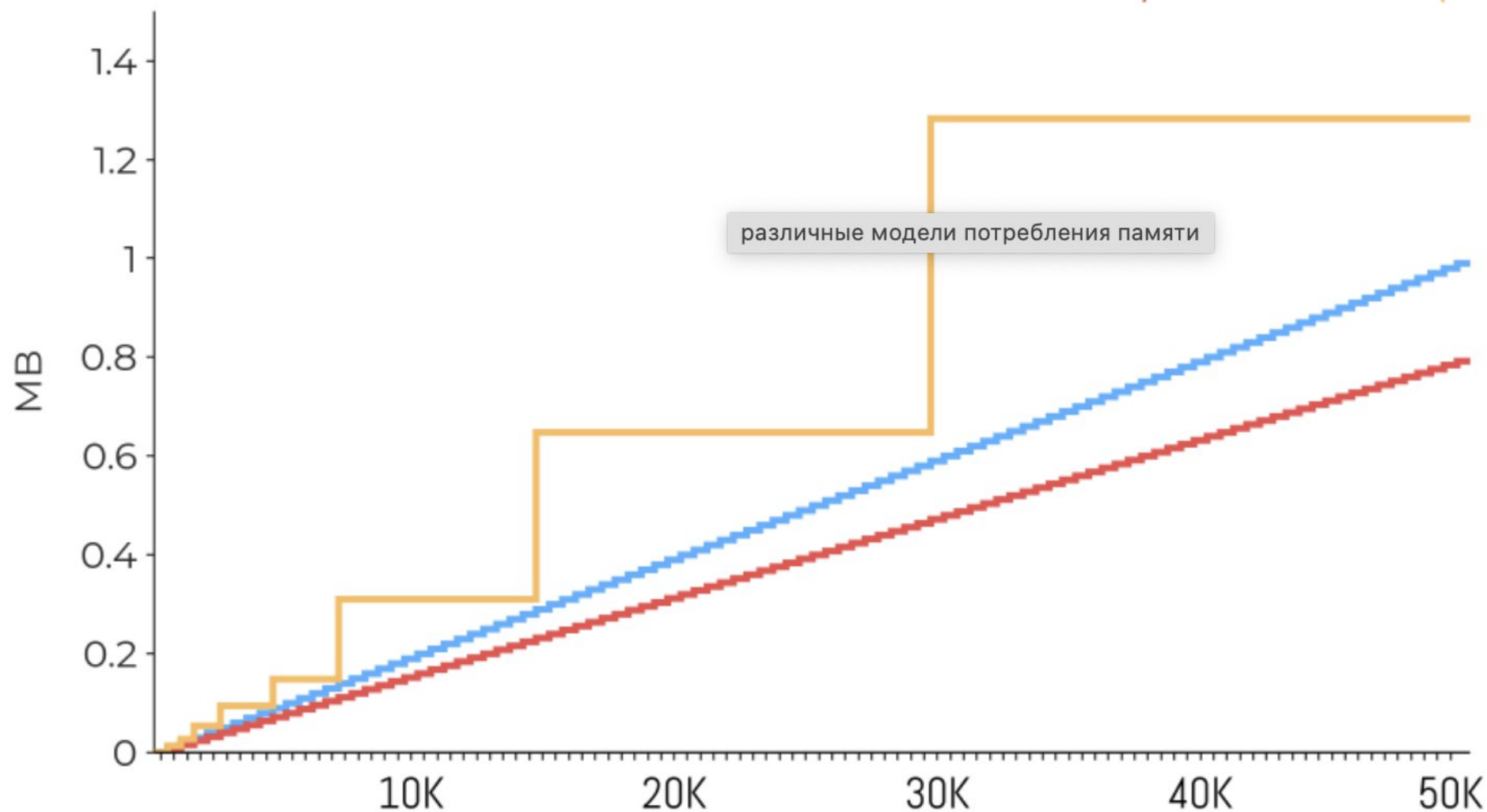


directory index (variable # of bits)

Преимущества

Chunk Index Memory Usage

— Swiss Table — Array — Builtin Map



различные модели потребления памяти

Потребление памяти стандартной реализации структуры данных `map` — выглядит как ступенчатая функция, поскольку она всегда создаётся с числом бакетов, равным степени двойки. Это связано с классической оптимизацией, основанной на побитовых операциях.

- Операции чтения и записи с хэш-таблицей более чем 1024 элементами стали быстрее примерно на 30%.
- Запись в заранее проалоцированную хэш-таблицу стала быстрее примерно на 35%.
- Итерация ускорила в среднем на 10%, а для хэш-таблиц с небольшим числом записей относительно размера — до 60%.

Вы можете отключить новую реализацию, установив **GOEXPERIMENT=noswissmap** во время сборки. Это актуально для старых процессоров в которых не реализован SIMD



ЭТОТ КОНТЕНТ КУПЛЕН НА САЙТЕ **SKLADCHIK.ORG**



Платформа

Это платформа, где каждый день тысячи людей собираются вместе, чтобы общими усилиями находить, приобретать и изучать курсы интересные именно им.



Сообщество

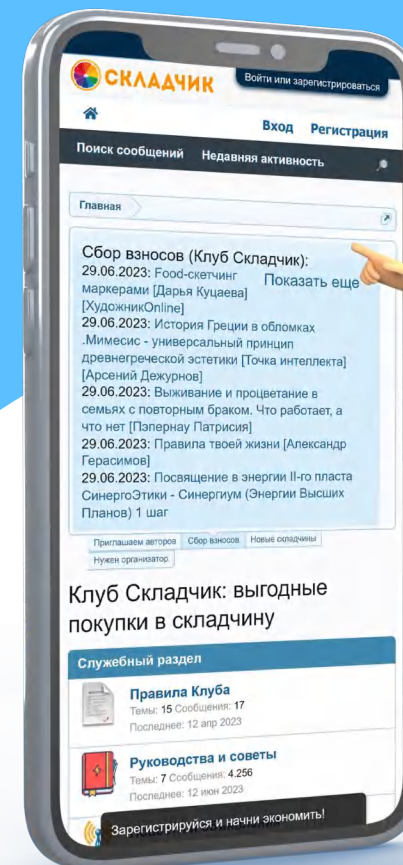
Это сообщество из 500 000 людей, открывших для себя выгодный способ быть в тренде самых актуальных и получать ценные знания по минимальной цене.



Библиотека

Это крупнейшая библиотека инфопродуктов в которой можно найти практически любой курс или тренинг продававшийся за последние 10 лет, а также уникальные авторские инфопродукты, которые не получится найти больше нигде.

ЧТО ТАКОЕ КЛУБ «СКЛАДЧИК»?



**ТЫ ЕЩЕ
НЕ В КЛУБЕ?**

**ПРИСОЕДИНЯЙСЯ
И НАЧИНАЙ ЭКОНОМИТЬ!**

