Хеш-таблица

- Данные хранятся в файле, который можно только дополнять (append-only log).
- В оперативной памяти хранится хеш-таблица (словарь).
- Ключ → Смещение (offset) в файле данных.

Хеш-таблица

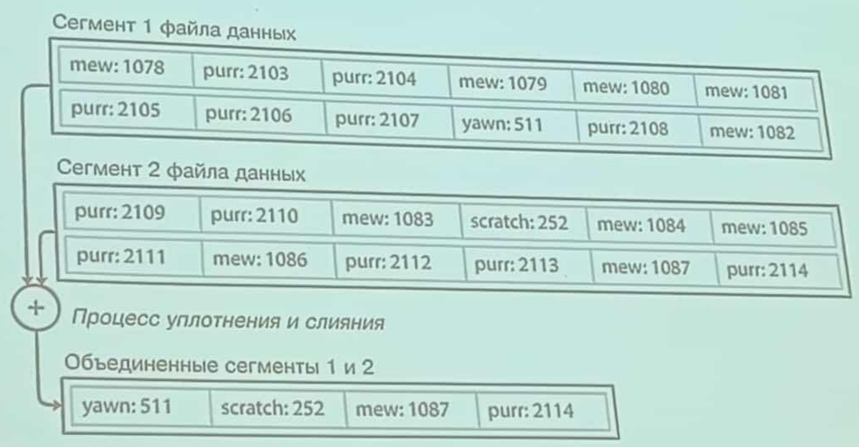
Сегменты

- Разбить журнал на сегменты фиксированного размера.
- При заполнении сегмента — закрыть его и начать новый.
- Запустить фоновый процесс уплотнения (compaction) и слияния (merging) сегментов.

Хеш-таблица
Слияние

- Объединение нескольких небольших сегментов в один новый.
- Позволяет уменьшить общее количество сегментов.

Хеш-таблица


Сегмент 1 файла данных
mew: 1078
purr: 2103
purr: 2104
mew: 1079
mew: 1080
mew: 1081
purr: 2105
purr: 2106
purr: 2107
yawn: 511
purr: 2108
mew: 1082
Сегмент 2 файла данных
purr: 2109
purr: 2110
mew: 1083
scratch: 252
mew: 1084
mew: 1085
purr: 2111
mew: 1086
purr: 2112
purr: 2113
mew: 1087
purr: 2114
Процесс уплотнения и слияния
Объединенные сегменты 1 и 2
yawn: 511
scratch: 252
mew: 1087
purr: 2114

Хеш-таблица
Ограничение

- Ключи должны помещаться в ОЗУ. Хеш-таблица на диске медленная из-за случайных чтений, дорогого расширения и сложности разрешения коллизий.
- Неэффективные запросы по диапазону. Невозможно быстро найти все ключи от 00000 до 99999. Необходимо обращаться к хеш-таблице для каждого ключа в диапазоне отдельно.

SS-table

Особенности

- Данные отсортированы по ключу
- Каждый ключ встречается только один раз (обеспечивается уплотнением)
- Порядок записи значений не важен - приоритет у более новых значений

Слияние

- Алгоритм, аналогичный сортировке слиянием
- Работает даже когда данные не помещаются в оперативной памяти
- При конфликте ключей берется значение из самого нового сегмента

Разряженный индекс

- Не нужно хранить все ключи - только некоторые ориентиры
- Пример: известны смещения для "handbag" и "handsome" → "handiwork" между ними
- Можно быстро просканировать небольшой диапазон
-

MemTable

- Сбалансированное дерево в оперативной памяти (красно-черное, AVL)
- Данные сохраняются отсортированными по ключу
- Быстрая вставка в любом порядке

Надежность

- Все операции немедленно записываются в журнал
- Журнал неупорядочен - используется только для восстановления
- После записи MemTable в SS-таблицу журнал удаляется

Алгоритм работы

- Запись: Добавляется в MemTable и параллельно пишется в журнал для надежности
- Чтение: Поиск в MemTable → последнем сегменте → предыдущем и т.д.
- Фоновые процессы: При превышении лимита MemTable записывается в SS-таблицу, Регулярное слияние и уплотнение сегментов

LSM-дерево

Фильтр Блума

- Эффективная проверка отсутствия ключа
- Избегание ненужных чтений с диска
- "Возможно есть" / "Точно нет"

Алгоритм работы

- Запись: Добавляется в MemTable и параллельно пишется в журнал для надежности
- Чтение: Поиск в MemTable → последнем сегменте → предыдущем и т.д.
- Фоновые процессы: При превышении лимита MemTable записывается в SS-таблицу, Регулярное слияние и уплотнение сегментов

Уплотнение

- Size-tiered: Объединение маленьких SS-таблиц в большие
- Leveled: Разбиение по уровням с постепенным перемещением
- Разные компромиссы между скоростью и использованием диска

B-Деревья

Опасность операций с множеством страниц

- Разбиение страницы при переполнении
- Запись двух новых страниц
- Обновление родительской страницы

Write-Ahead Log

- Файл только для добавления (append-only)
- Быстрая последовательная запись
- Используется для восстановления после сбоев

Защелки (latches)

- Облегченные версии блокировок
- Защита структур данных при одновременном доступе
- Высокая производительность

- LSM-деревья: Обычно быстрее при записи
- B-деревья: Обычно быстрее при чтении