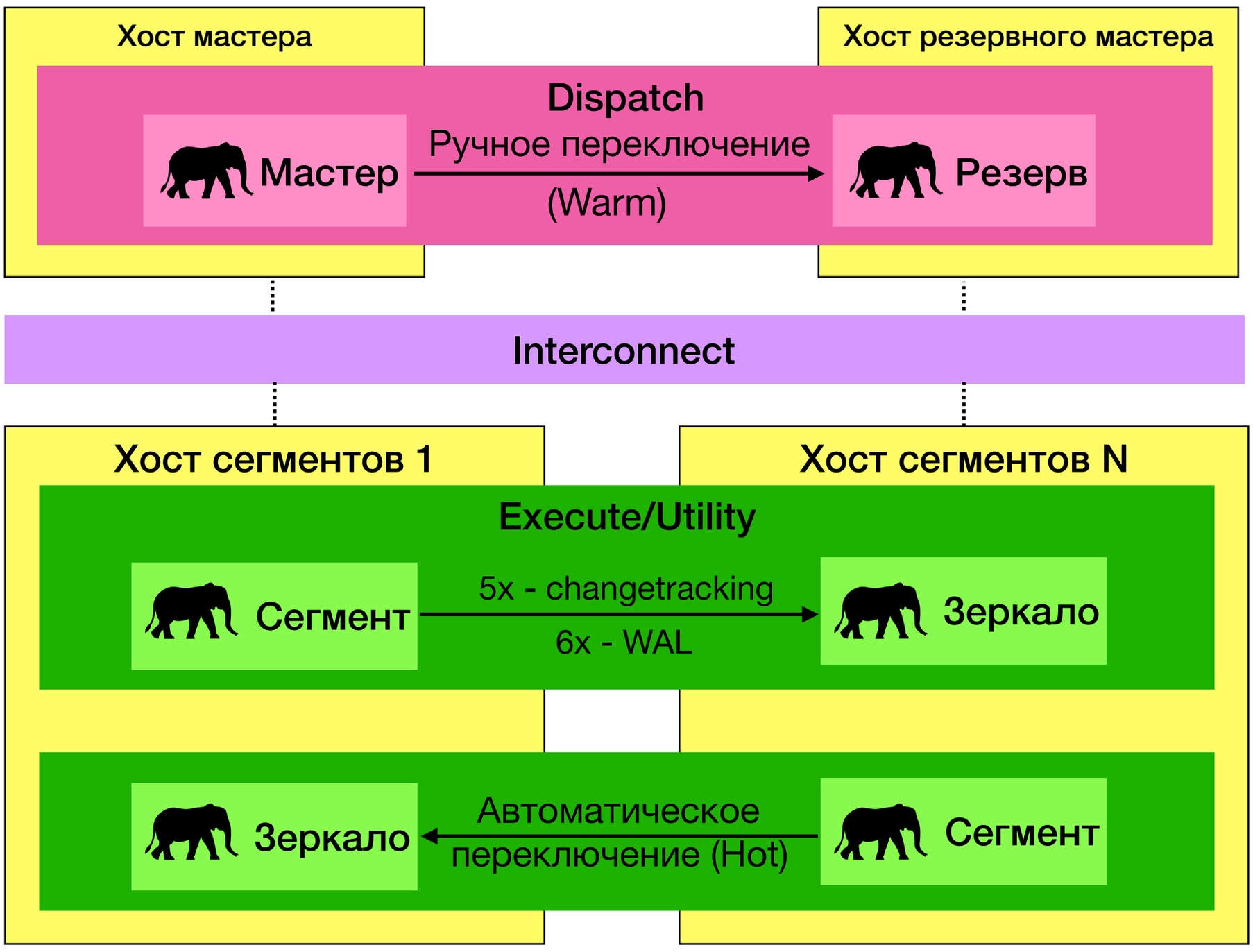
Greenplum

Внутреннее устройство PostgreSQL для аналитики

Что такое Greenplum?

- PostgreSQL (5x - 8.2, 6x - 9.4)
- Кластер (MPP)
- Полиморфное хранилище
- Сетевой протокол
- Разграничение ресурсов
- Движок хранения для аналитической нагрузки

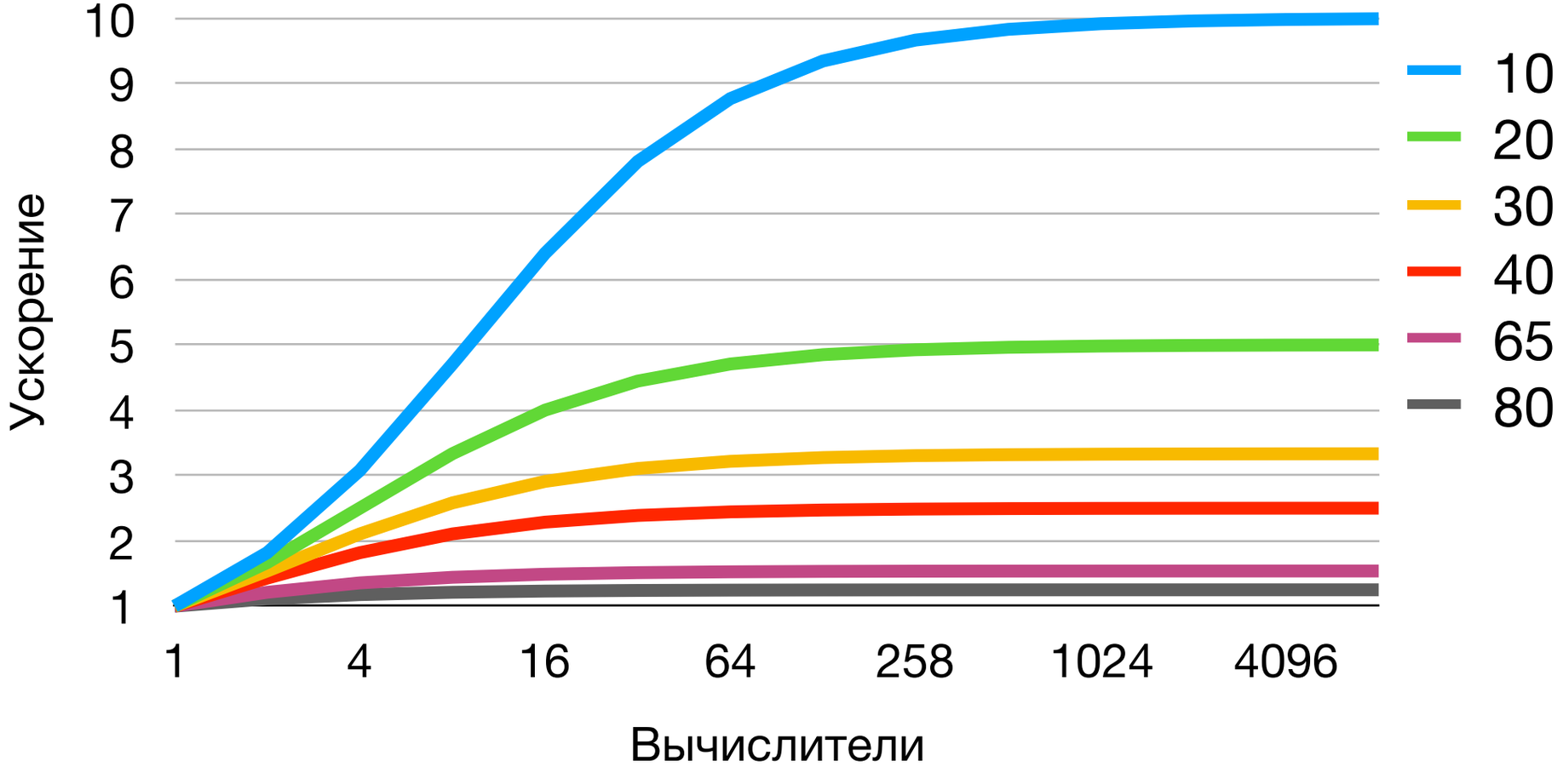
Схема кластера



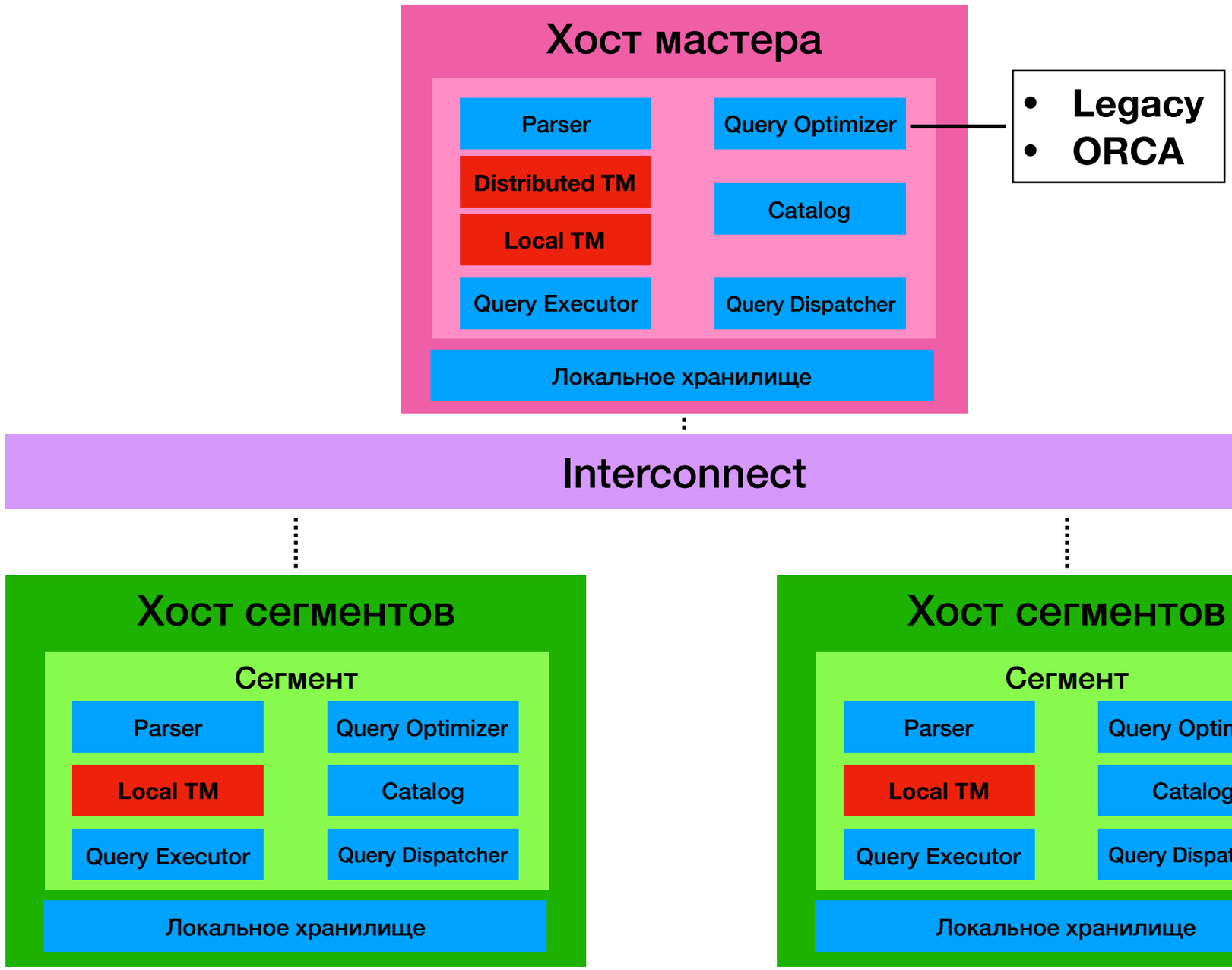
Закон Амдала

$$S_p = \frac{1}{a + \frac{1-a}{p}}$$

a - доля последовательных вычислений
p - количество вычислителей
S - ускорение

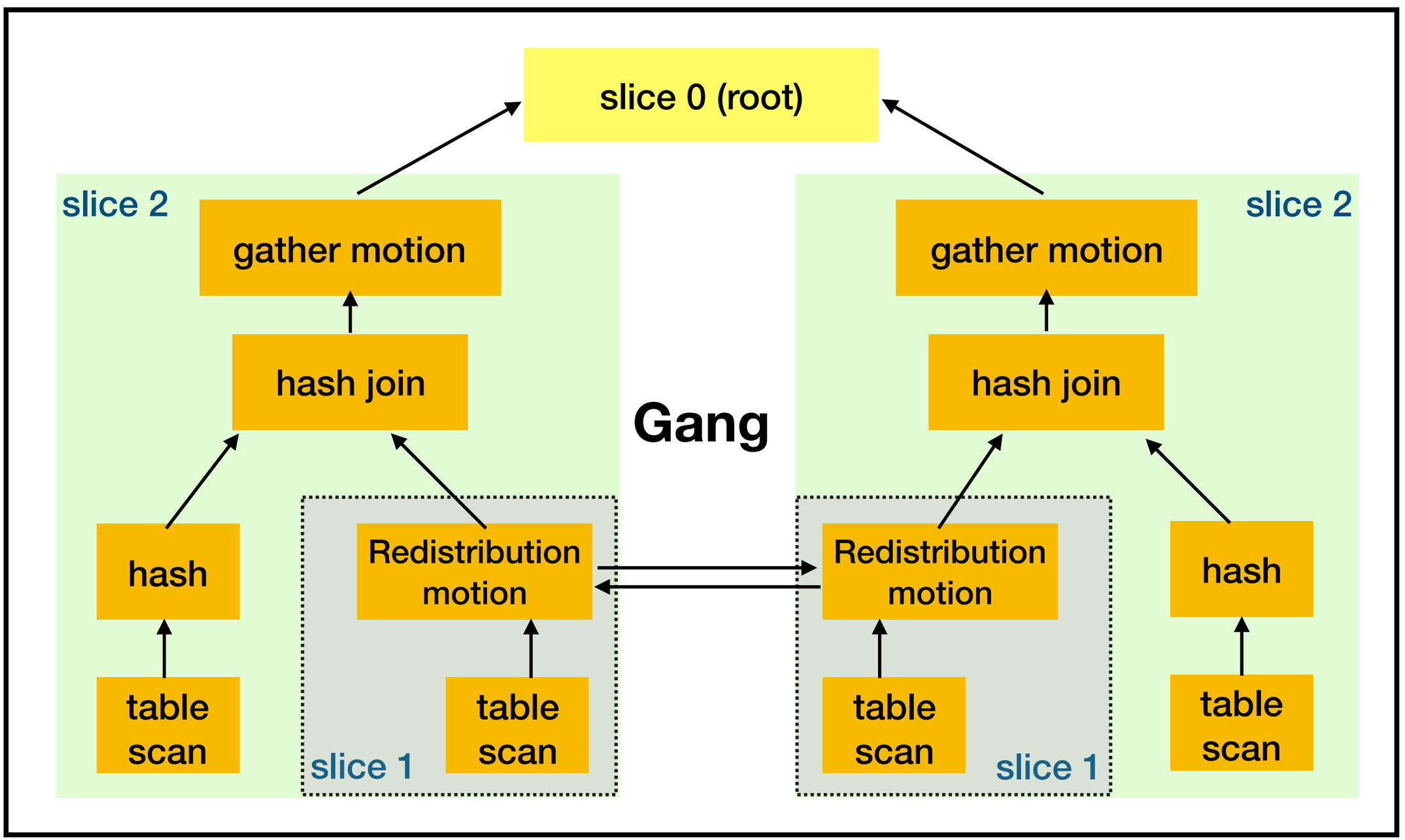


Распределенные транзакции

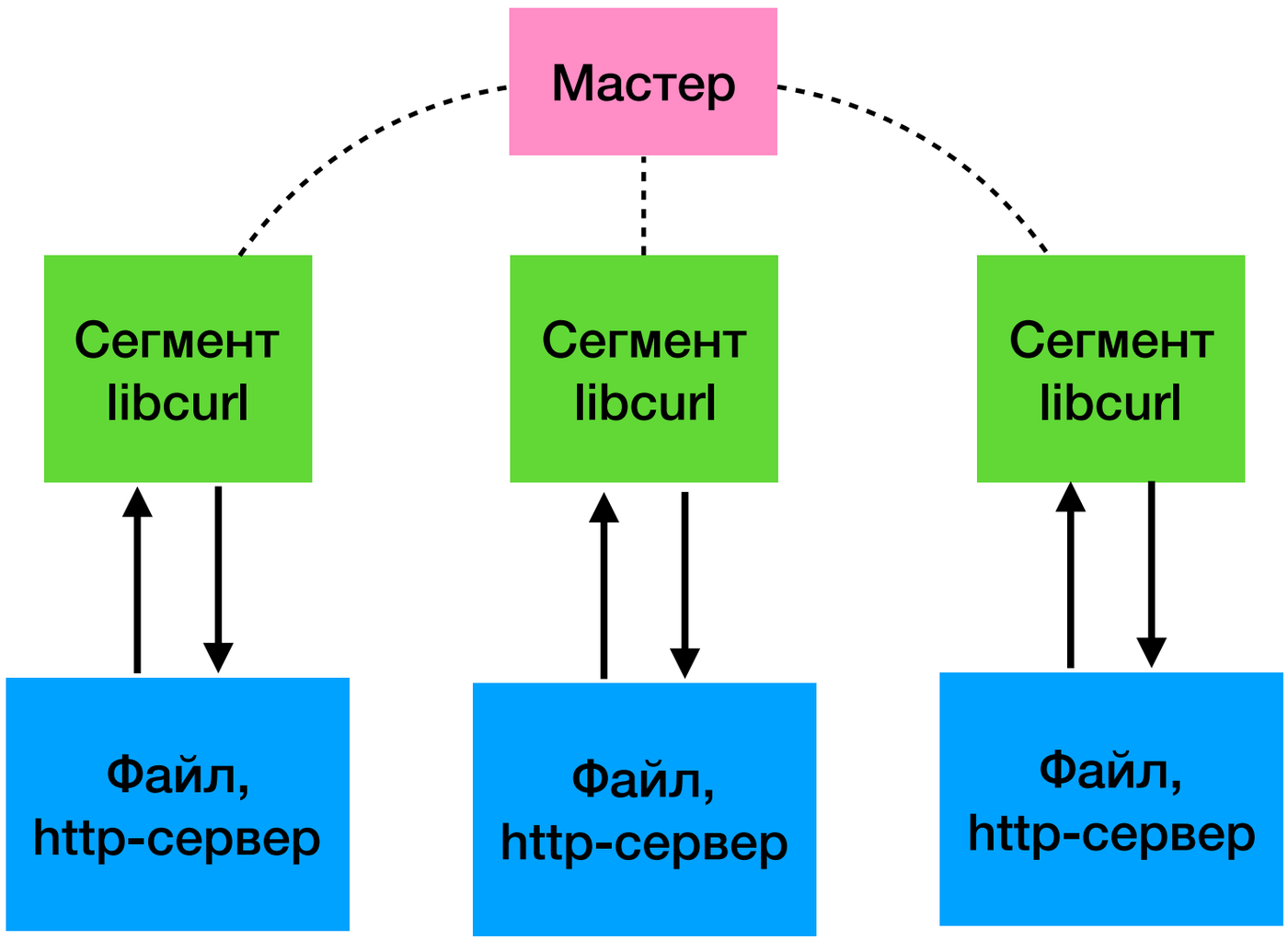


Распределенные запросы

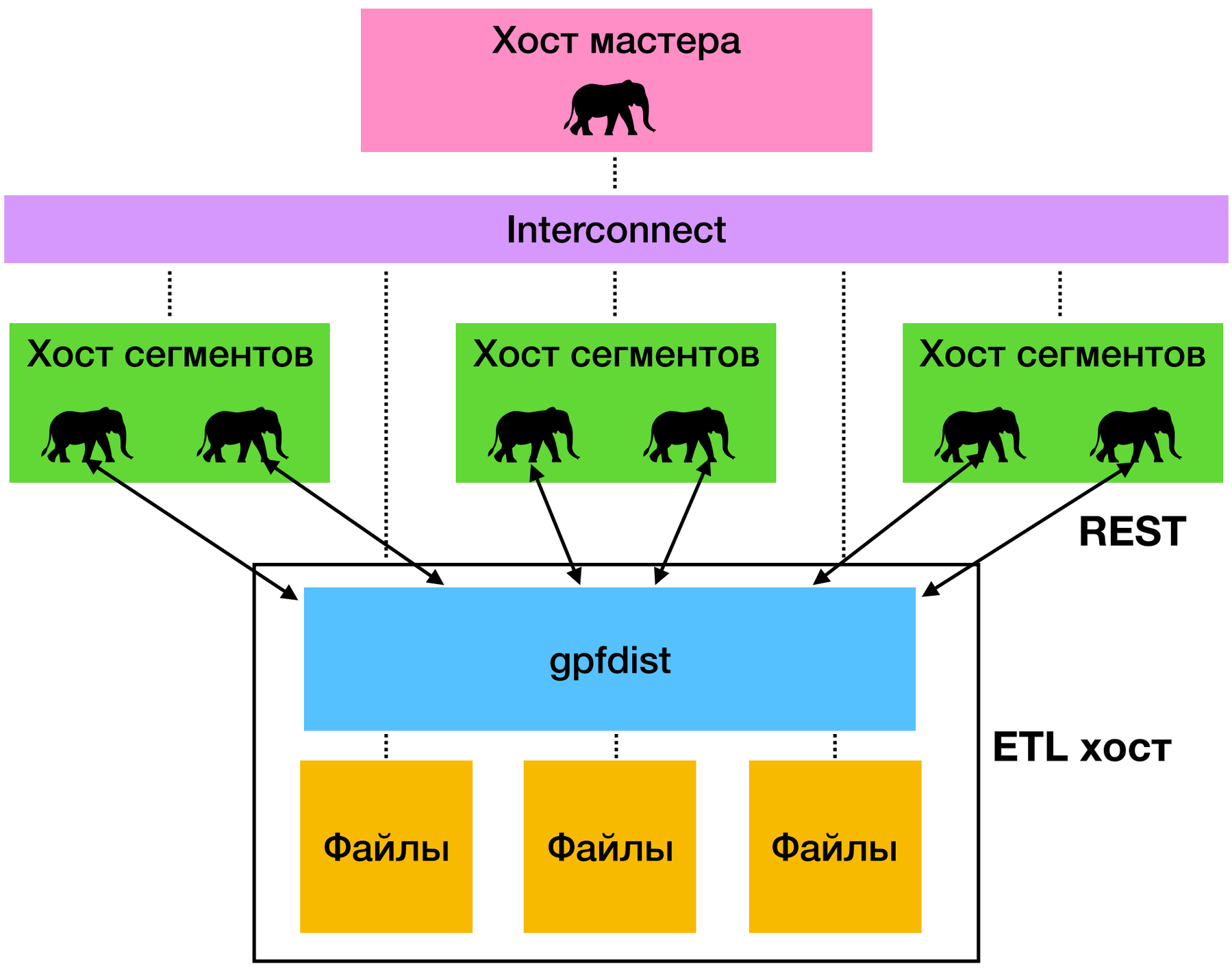
select t1.id, t2.val from t1 join t2 using(val) where t1.cond = 1;



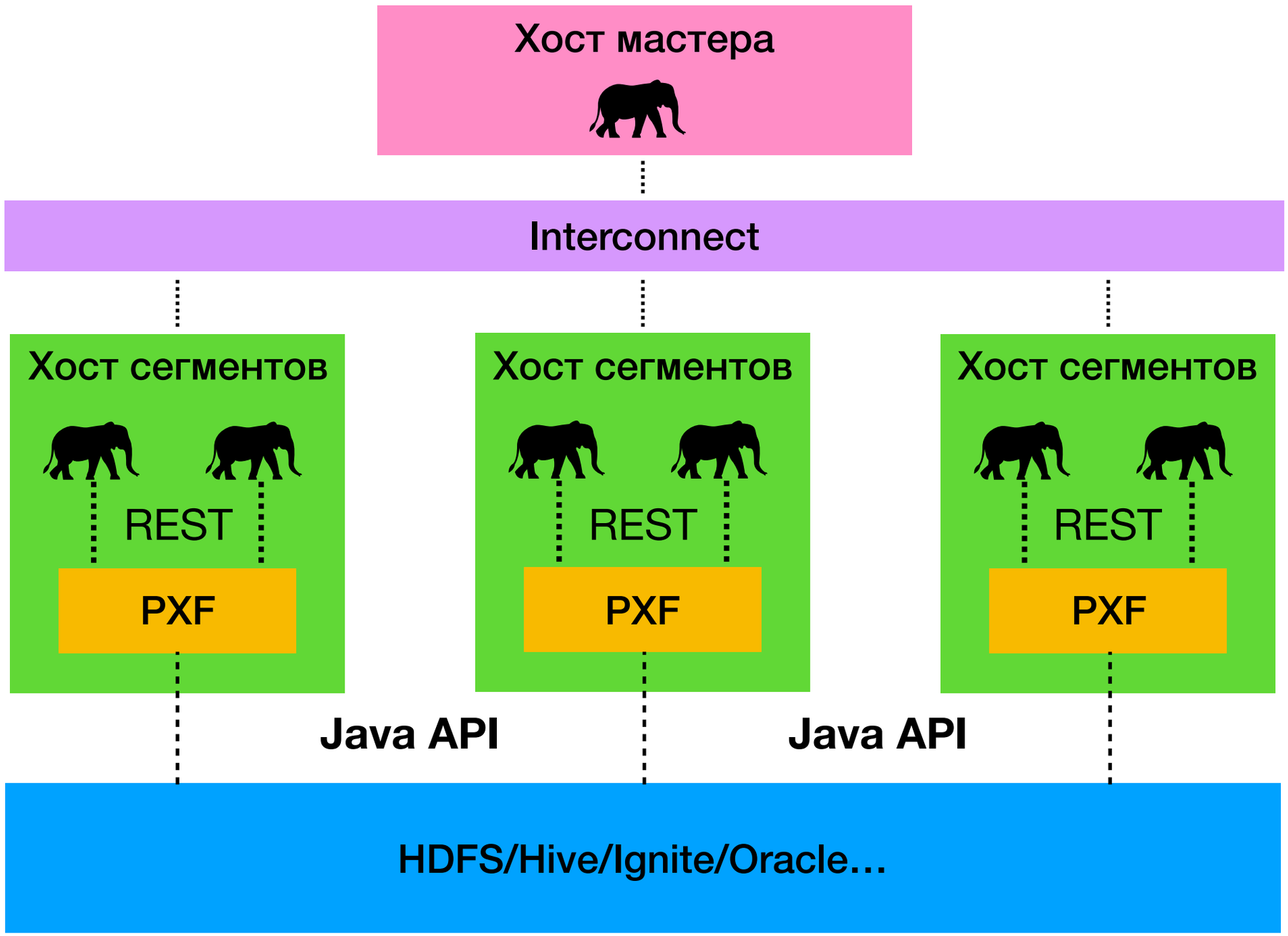
Параллельная загрузка и выгрузка



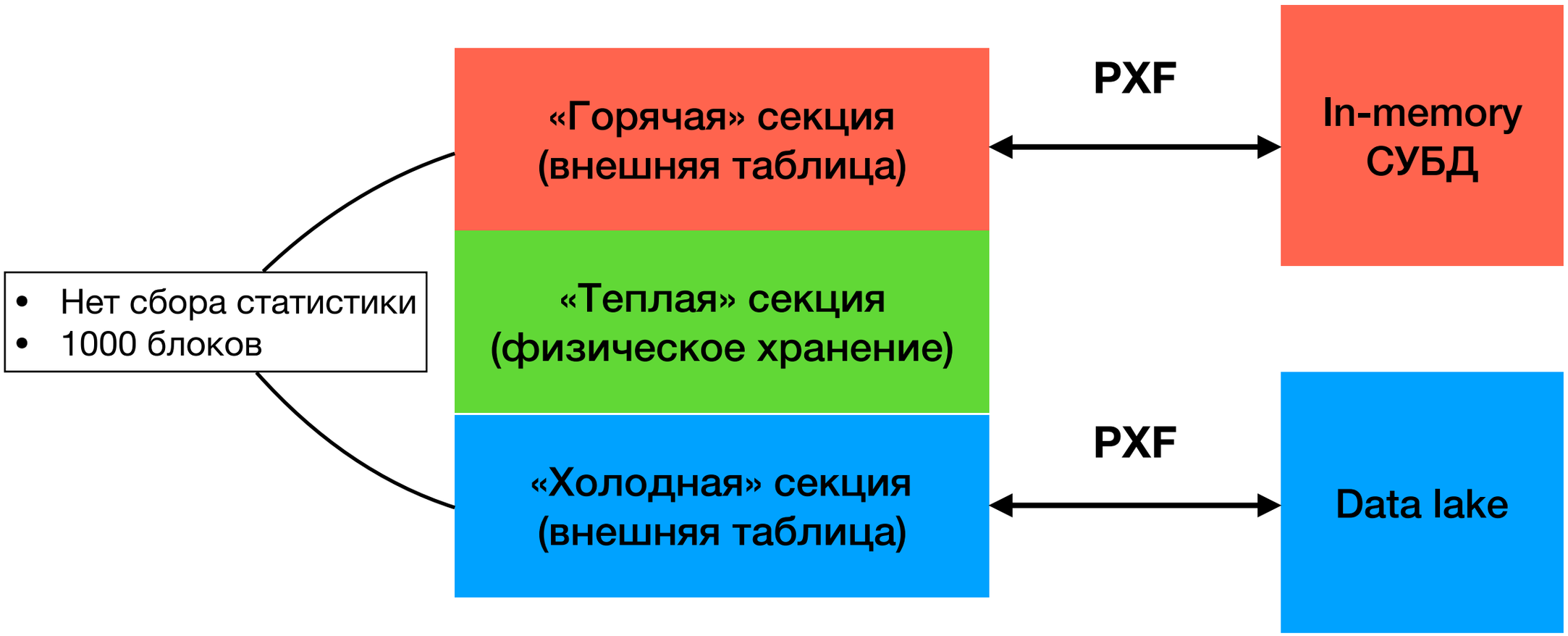
gpfdist


Хост мастера
Interconnect
Хост сегментов
Хост сегментов
Хост сегментов
REST
gpfdist
ETL хост
Файлы
Файлы
Файлы

Platform Extension Framework



Полиморфное хранилище



Сетевой протокол

- Interconnect
  - UDPIFC (UDP + контроль потока)
  - TCP
    - Latency - задержки при установлении соединений
    - Throughput - соизмеримо с UDPIFC
- Проблема нехватки портов в больших кластерах (200 сегментов на 30 хостах)

Разграничение ресурсов

- Ресурсные очереди (устарели)
- Ресурсные группы (cgroups)
  - ЦПУ (полоса/ядра)
  - Память (лимиты, сброс на диск)
  - Количество параллельных транзакций

Аналитическая нагрузка

Профиль:

- «Широкие» таблицы (сотни столбцов, много строк)
- Работа большими пакетами данных (чтения, вставки)
- При чтении обрабатывается много строк, но мало столбцов (1-10%).
- Результат выполнения запроса мал по сравнению с обработанными данными (итог агрегируется или фильтруется)
- Обновления и удаления редки
- Мало параллельных запросов в системе
- Допустимы задержки при получении результата

Цель: максимизация пропускной способности запроса

Уменьшаем I/O

- Движок хранения
  - Метаданные отдельно от значений (есть)
  - Столбцы (есть)
    - Сжатие (есть)
    - Сортированные блоки (нет)
    - Разреженные индексы (нет)
- Секционирование (есть)
- Сегментирование (есть)

Метаданные

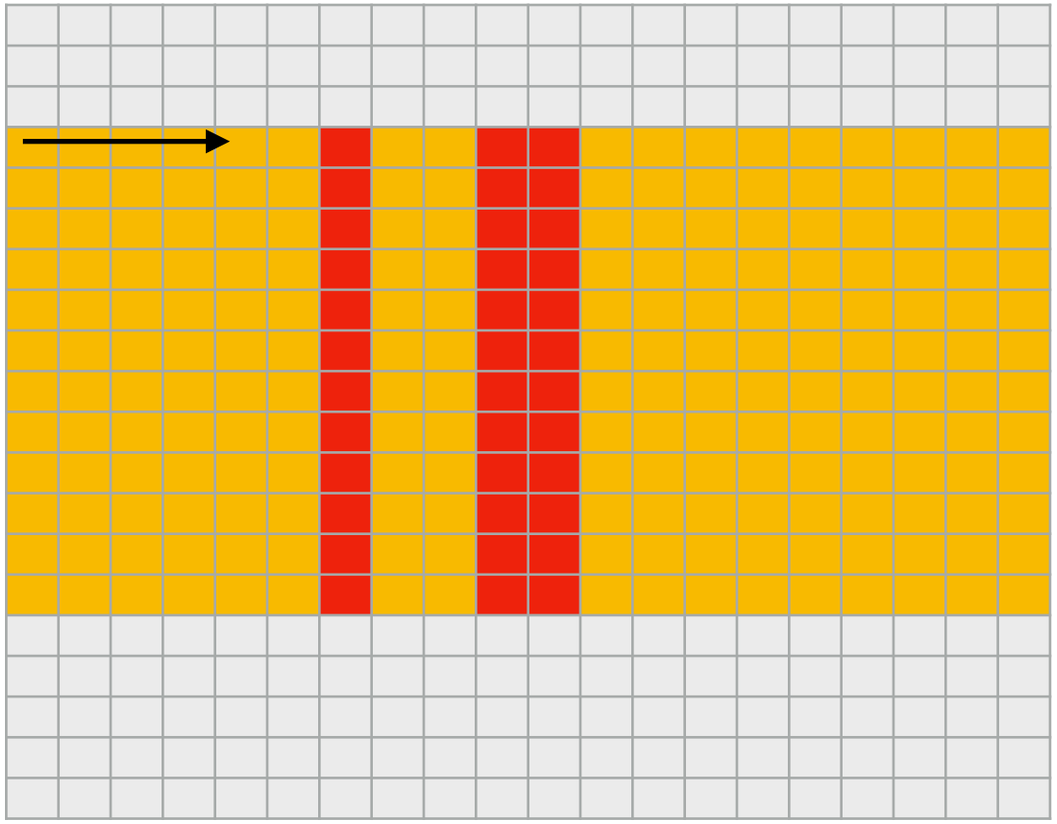
MVCC + hint bits

xmin	xmax	…	DATA
100	0	…	1,AAAA
100	0	…	2,BBBB
100	0	…	3,CCCC
100	0	…	4,DDDD
…	…	…	…

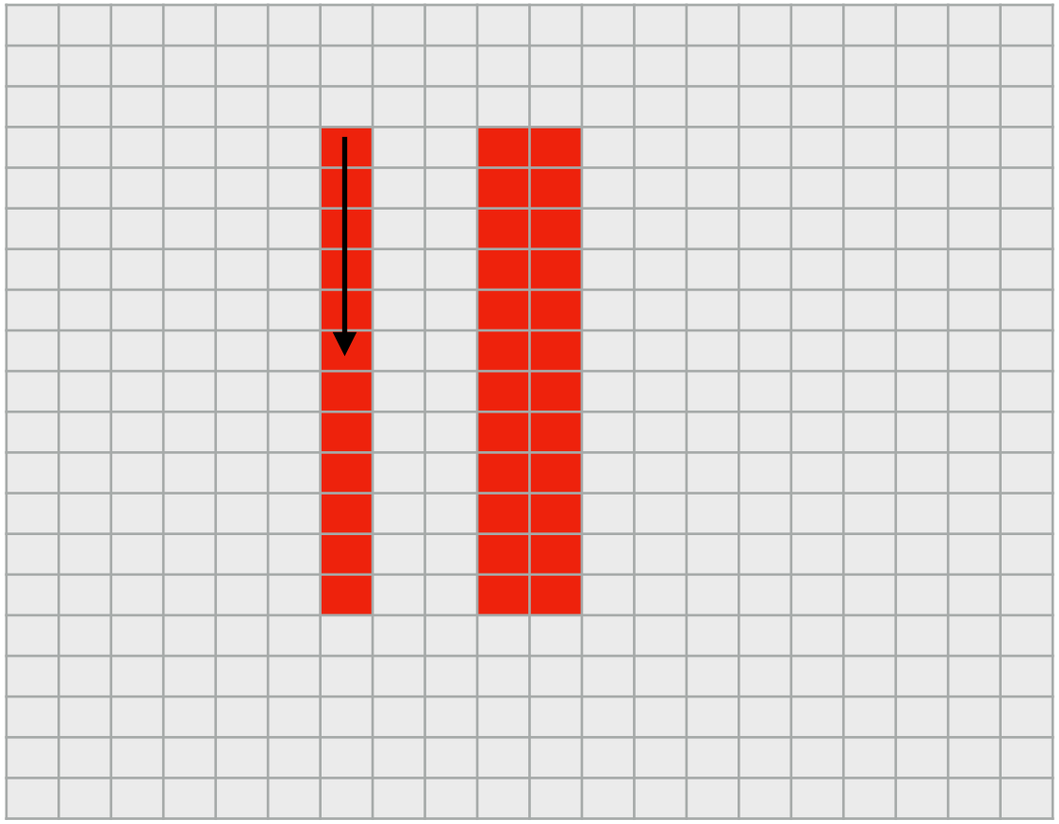
При вставке большими пакетами метаданные идентичны

Столбцы

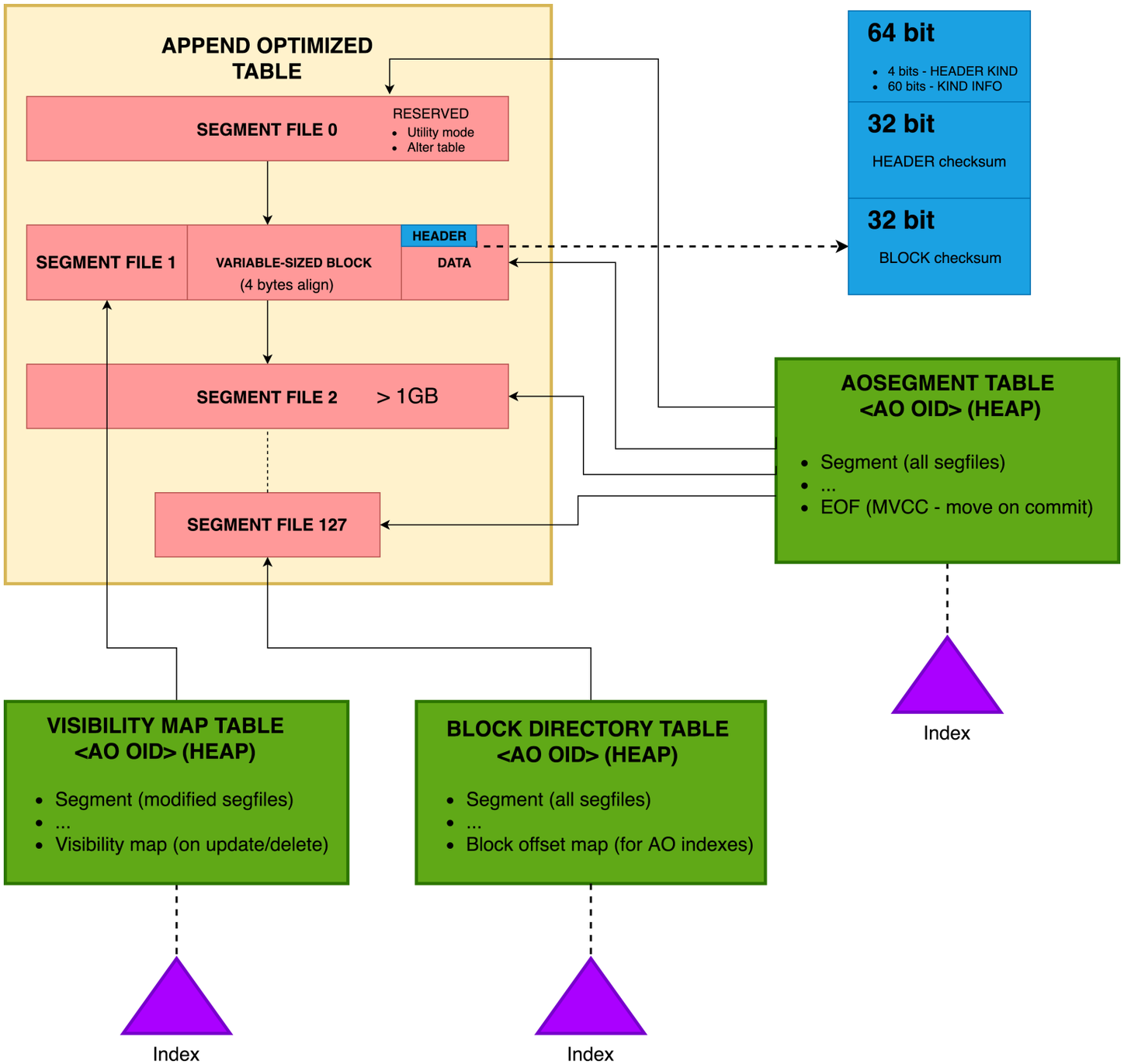
Строковое хранение



Столбцовое хранение

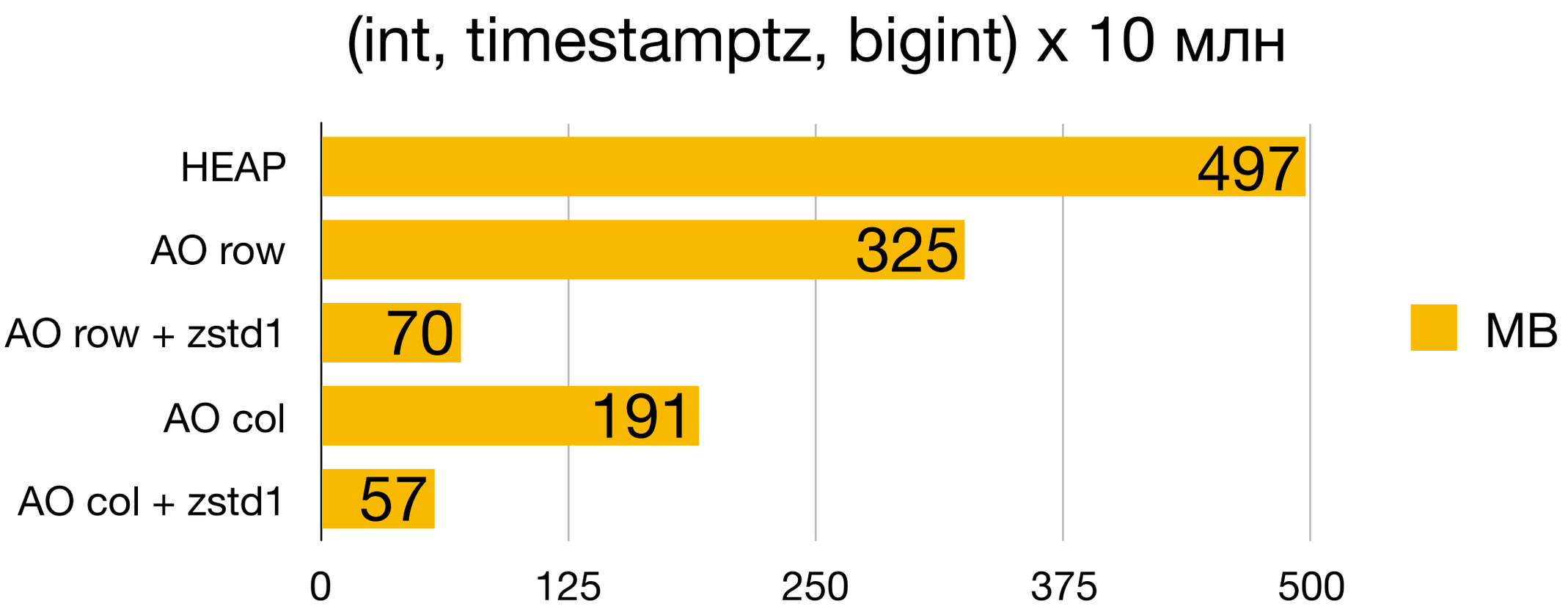


AO-строки/столбцы



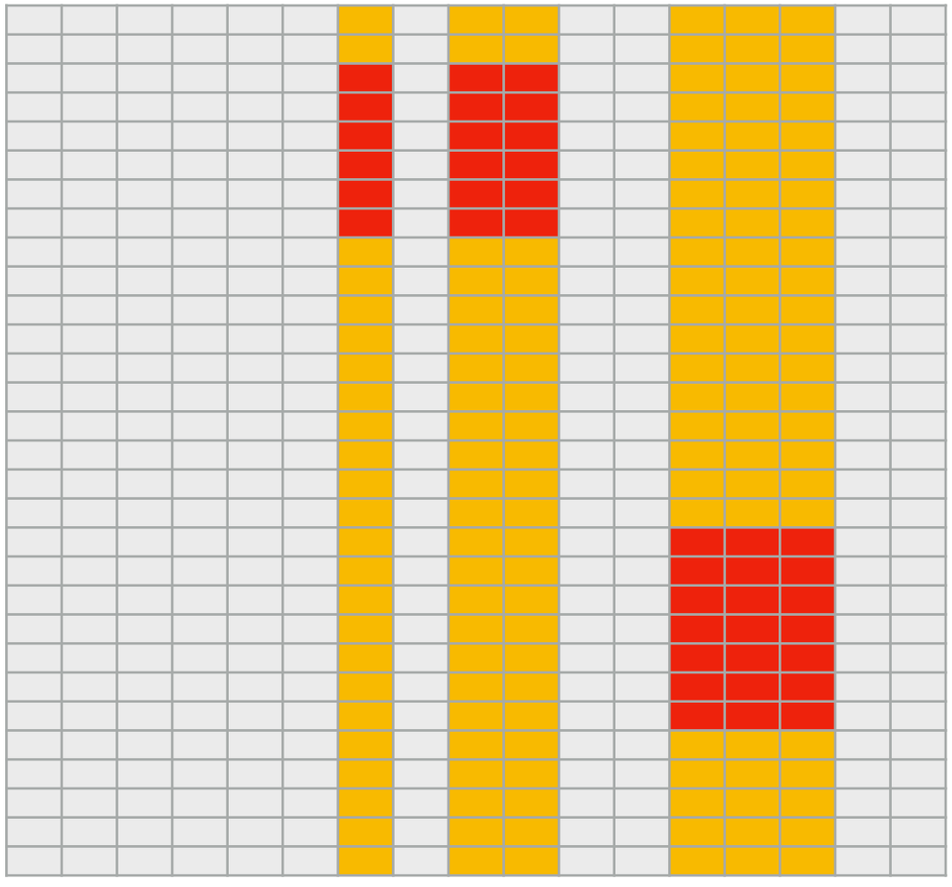
Сжатие

- **zlib (1-9)**
- **zstd (1-9)**

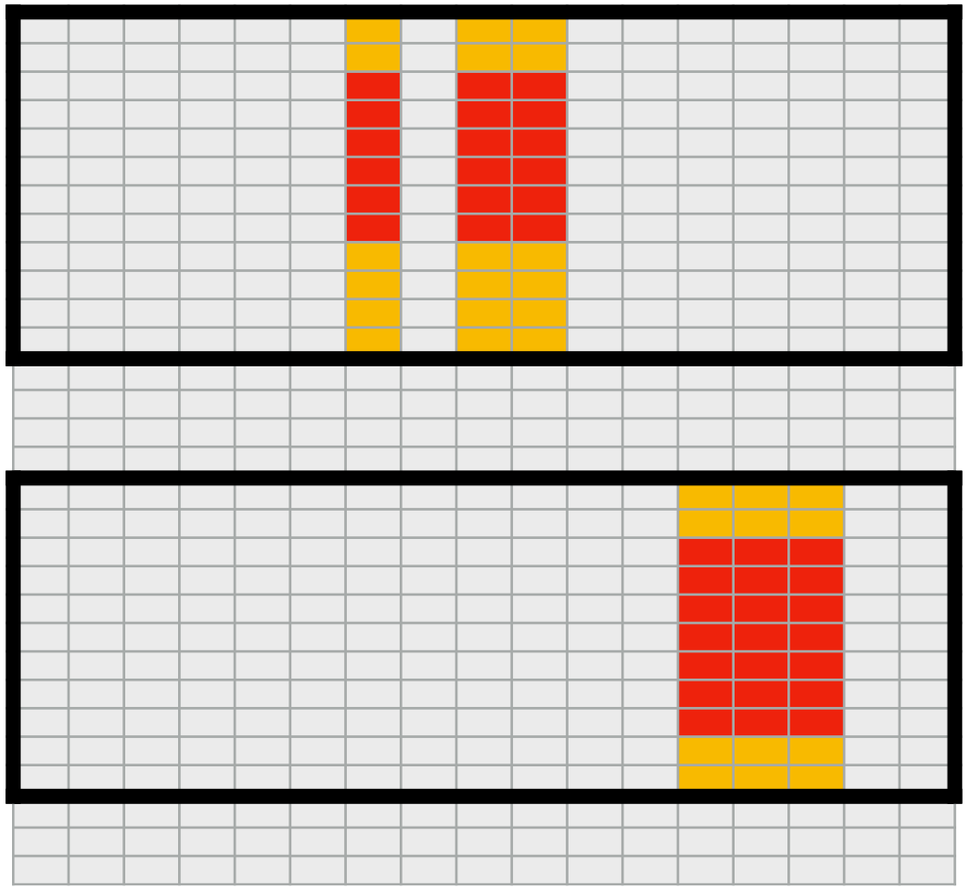


Секционирование

Без



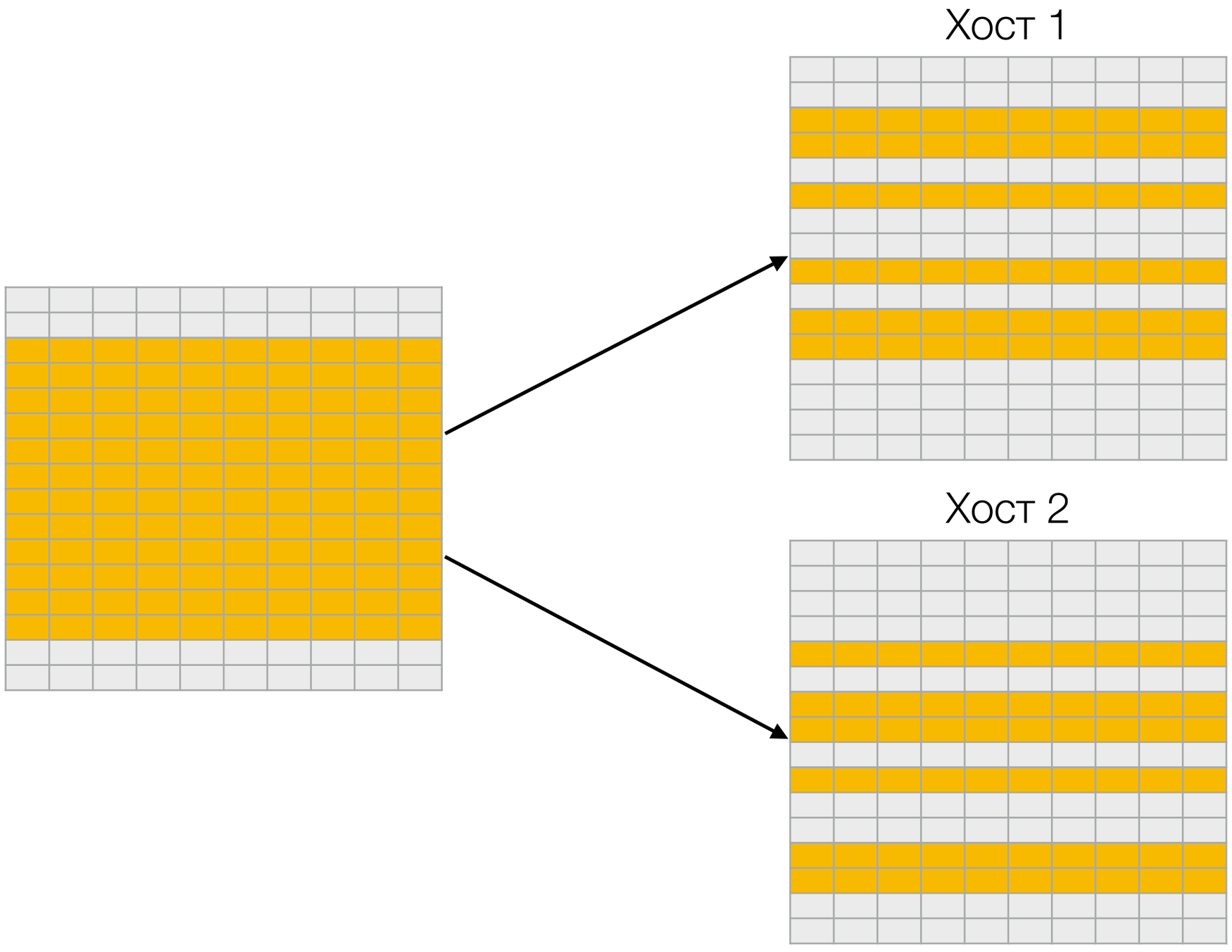
С



Секционирование в GP

- Виды секций
  - Диапазон
  - Список
- Уровни вложенности
  - Быстрый рост количества секций и файлов (столбцы)
  - На одном уровне не более 32767 секций

Сегментирование



Сегментирование в GP

- Политики сегментирования
  - 5x - случайное (round-robin)
  - 5x - по ключу
  - 6x - реплицированное
- Алгоритм хеширования
  - 5x - FNV1_32 (SIMD)
  - 6x - JUMP (консистентное хеширование)

Проблемы распределенной сборки мусора

Несогласованный autovacuum

- Различные значения локальных счетчиков транзакций на сегментах
- Непредсказуемость выполнения в кластере
- Время запроса определяется самым медленным сегментом

Падение скорости распределенного запроса до **2** раз

Сборка мусора в GP

- Autovacuum отключен (исключение template0)
  - Мало транзакций
  - Нагрузка на чтение (виртуальные XID)
    - Добавление и модификация осуществляются большими пачками данных
- Согласованный vacuum/freeze
  - Единовременное выполнение по всему кластеру
  - Индивидуальная реализация расписания (привязка к ETL)

Присоединяйтесь!

https://t.me/joinchat/DYEBbRGynrZ0dL-QaKmkbw

Благодарю за внимание

https://habr.com/users/darthunix/
telegram: @darthunix