



Data Pipeline 3.0

Обзор Tarantool

Александр Краснощёков



@mail.ru
group

Что такое Tarantool

Где используется?



- Очень быстрая СУБД с сервером приложений lua / swift / c
- Открытый код, BSD, есть компоненты closed source
- Способность обрабатывать огромное кол-во одновременных подключений
- Соответствие ACID
- Отказоустойчивость и масштабируемость (репликация, шардинг)
- Кооперативная многозадачность, неблокирующие операции IO (включая работу с внешними сервисами и файловой системой)

- Смешанная модель данных
 - Поддерживается реляционная модель хранения: таблицы (спейсы),
 - Поддерживаются документы
- Разнообразные индексы
 - По типу: hash, tree, r-tree, ...
 - По назначению: первичные и вторичные
 - Составные
- Поддержка SQL (beta)

- Движок memtx
 - In-memory с надежной записью на диск (snapshot + WAL)
синхронная / асинхронная запись в лог
 - Только in-memory
- Движок vinyl
 - Дисковый движок на LSM-деревьях
 - Может использоваться, когда данные не вмещаются в RAM
 - Кеш в памяти
 - Оптимизирован на работу с дисками
 - Эффективен на больших объемах данных

- Как с классической СУБД, используя коннекторы
 - Тарантул использует бинарный протокол
 - Существуют коннекторы для большинства языков: Java, python, c, c#, php, perl, js, erlang, R ...
<https://tarantool.io/en/doc/1.9/book/connectors/index.html>
 - Для достижения максимальной производительности - batching
- Как с сервером приложений
 - В этом случае можно использовать транзакции
 - Характерно отсутствие latency – для ряда задач дает выигрыш в производительности на несколько порядков
 - Можно с помощью nginx превратить тарантул в REST / http RPC – сервис

Для каких задач стоит применять Тарантул

- Для задач, где:
 - ...важно время обработки запроса
 - ...данные часто устаревают, работа с 'горячими' данными (например, OLTP)
 - ...необходимо устранить latency (например, для обработки одной операции необходимо сделать много последовательных запросов)
- Паттерны:
 - Транзакционный кеш, OLTP
 - Realtime-аналитика, расчет предикторов
 - ETL
 - Очереди, сессии, авторизация, ...
 - Антифрод, антибрутфорс
 - Таргетинг
 - IoT

Для каких задач не стоит применять Тарантул

- Тарантул не эффективен в качестве аналитической СУБД

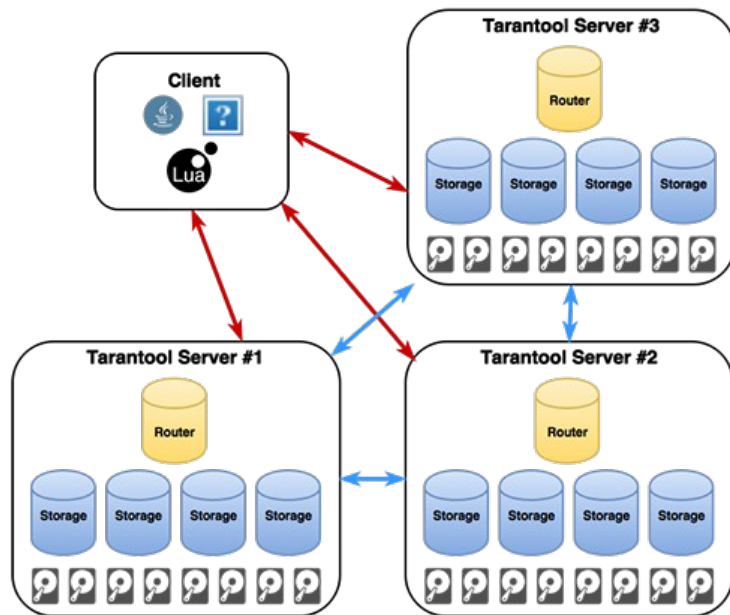
Для этих задач следует использовать DWH, колоночные распределенные решения



- Для Tarantool существует много модулей, созданных как сотрудниками mail.ru, так и community
Список модулей на странице <http://tarantool.io>
- В тарантуле могут использоваться c-библиотеки, подключенные через ffi
- Коннекторы к внешним системам:
 - СУБД: mysql, PG, Oracle, ...
 - Обращение к внешним сервисам по http (REST)
 - Работа с файловой системой

За счет чего достигается оптимальность

- Небольшой memory footprint
- Один рабочий поток упрощает алгоритмы работы тарантула, позволяет использовать меньше памяти для мета-информации и системных объектов
- Эффективные cache-agnostic алгоритмы
- Возможность использования мультиплексирования (pipelining, batching)
- Возможность писать код приложения рядом с данными (внутри инстанса БД)



- Репликация
 - Синхронная (выборочно) / асинхронная
- Шардинг