Rubuki

Социальная сеть, которая просто работает

В начале

- Нерабочая схема.
- Отсутствие готовой функциональности.
- Жуткие тормоза в базе.
- MySQL.

Что делать?

– Нет.


Мы пойдем
другим путем

Наш ответ

Сложные задачи и их решения

- Рекомендательный сервис
- Система премирования
- Полнотекстовый поиск

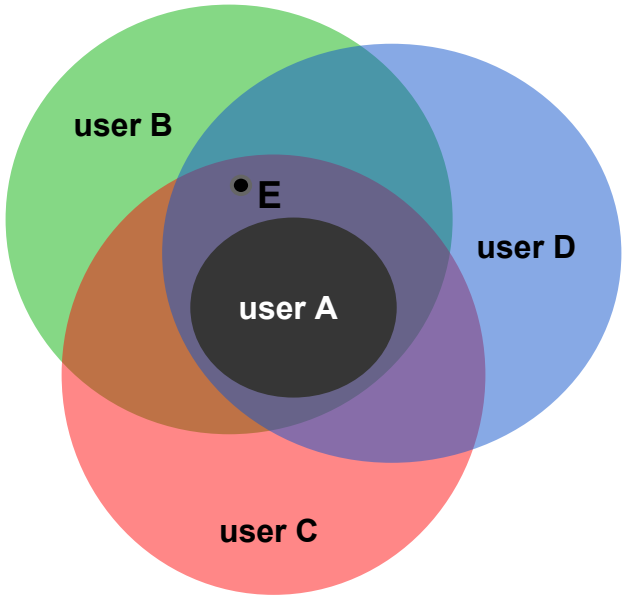
Рекомендательный сервис

Основные требования

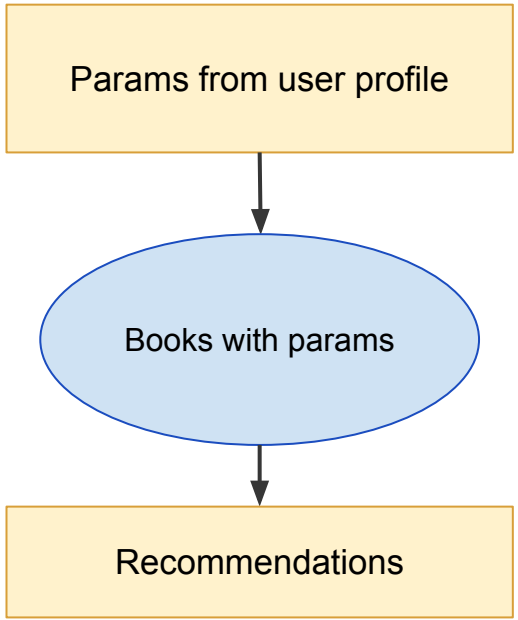
- Обновлять рекомендации не реже раза в сутки.
- Минимально возможная нагрузка на сервер.
- Динамичность рекомендаций для каждого пользователя.
- Процент вероятности, что рекомендация близка пользователю.
- Дополнительный расчет рекомендаций, по данным из профиля.

Решение

$R_{рек} = (S_{Ub} + S_{Uc} + S_{Ud}) - S_{Ua}$; $E_{count} = 3$;



+



Анализ решения на PostgreSQL

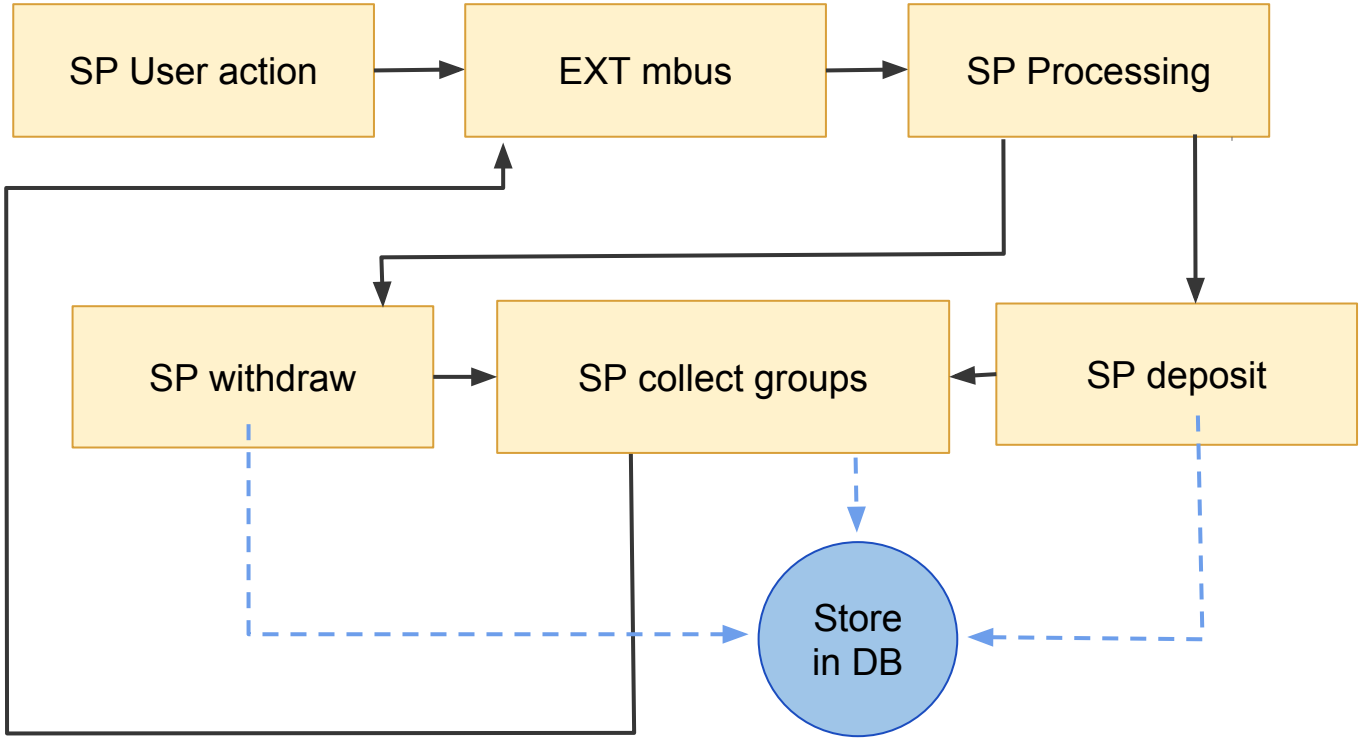
- Скорость работы в пределах 100мс на пользователя.
- Возможность распределять выполнение по времени.
- Динамический поиск вхождений.
- Подсчет процента вероятности, с которой пользователю понравится рекомендация.
- Расчет дополнительных рекомендаций на основе данных из профиля пользователя.

Система премирования

Основные требования

- Транзакционность.
- Возможность ручного начисления.
- Автономное формирование групповых начислений, по результатам накопления групп.
- Обновление баланса в режиме реального времени.
- История начислений.
- Распределение нагрузки.

Решение на PostgreSQL



Анализ работы решения

- Отказоустойчивость.
- Асинхронная работа (отсутствие нагрузки).
- Легкий рефакторинг логики.
- Повторное использование атомарных операций.
- Реализованы все требования.

Полнотекстовый поиск

Требования

- Быстрый поиск данных.
- Возможность масштабирования.
- Ранжирование результатов.
- Поиск по разным критериям.
- Высокая скорость индексации.

Противопоставление технологий

Обычно

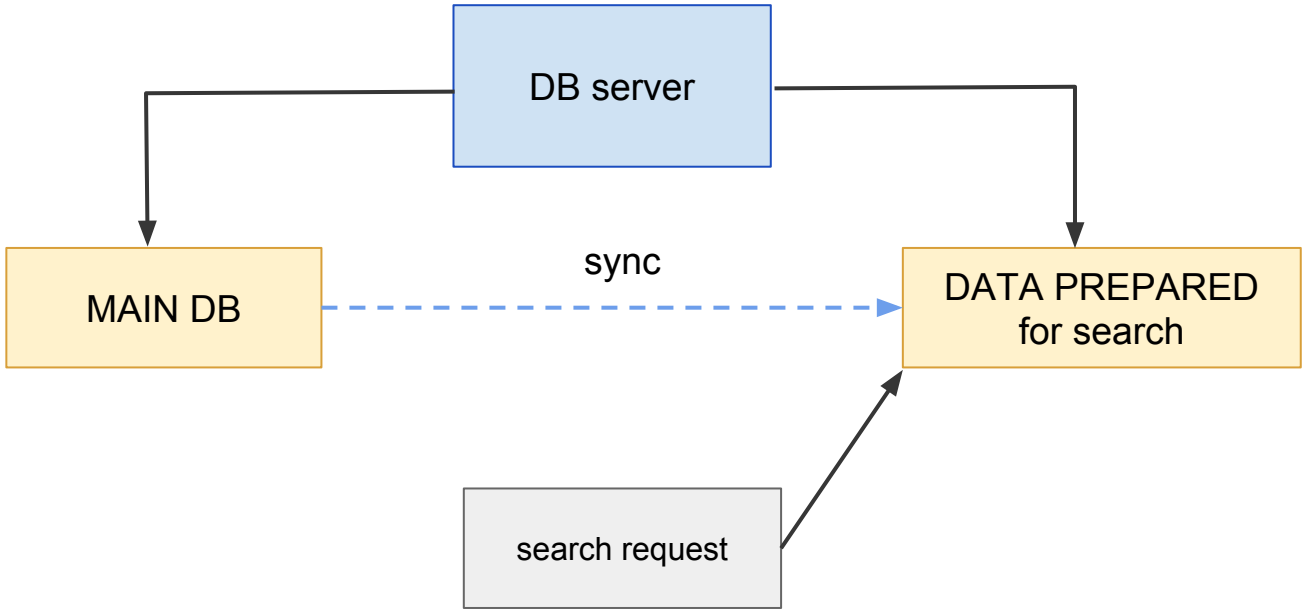
У нас

Почему PostgreSQL?

- Стабильность решения.
- Возможность масштабирования.
- Приемлемая скорость работы.
- Ожидаемая нагрузка на сервер.
- А почему бы и нет?

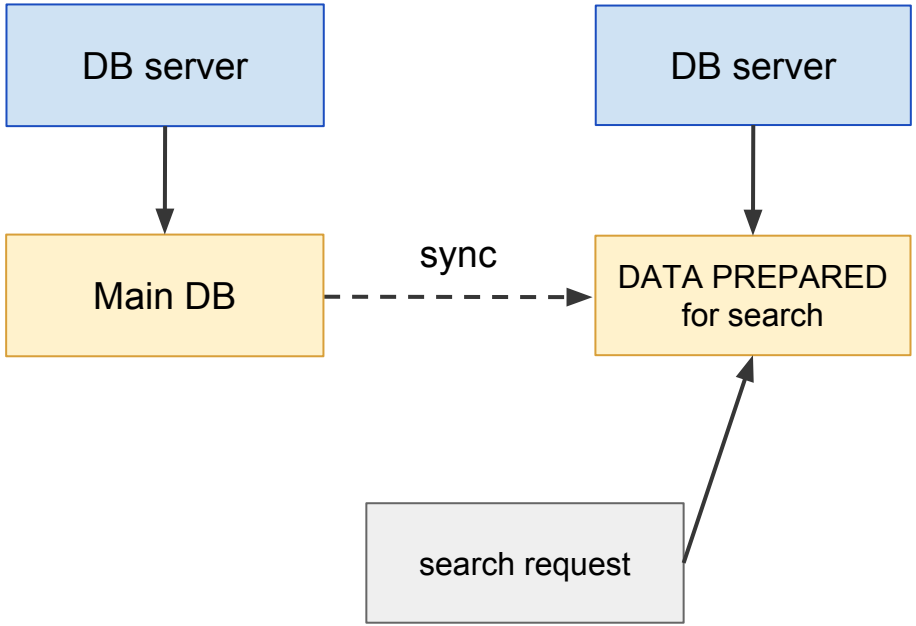
Схема работы

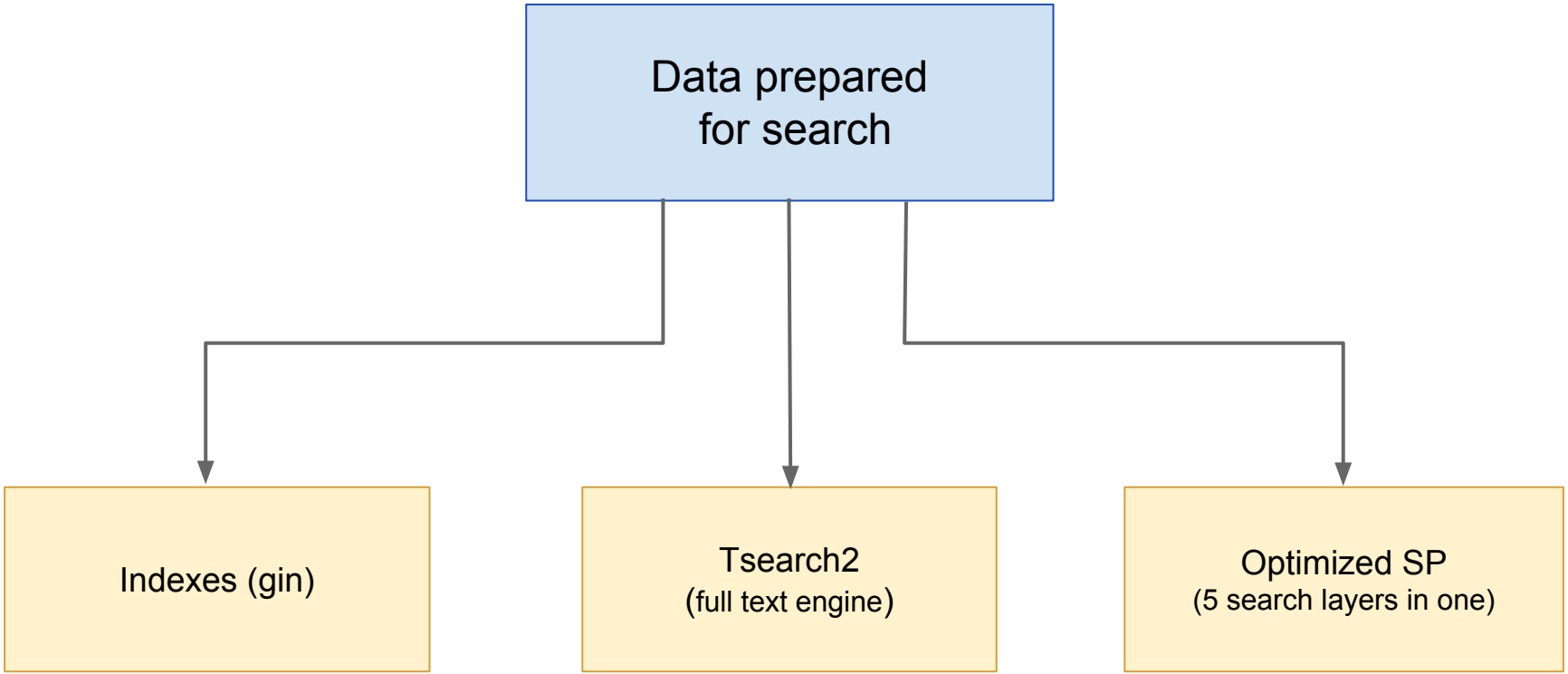
На начальном этапе развития проекта



Как может работать?

При необходимости




Data prepared
for search
Indexes (gin)
Tsearch2
(full text engine)
Optimized SP
(5 search layers in one)

Логика в базе

Плюсы

- Скорость работы.
- Быстрое внесение изменений.
- Отсутствие неконтролируемого роста запросов.
- Транзакционность.
- Встроенные решения конкурентного доступа.

Минусы

- Незначительные.

Ваши вопросы?