

Никита Митцев

Разработка ограничителя частоты запросов для Яндекс.Трекера

Руководитель проекта: Эдуард Александров

23.12.2002



Факультет математики и компьютерных наук СПбГУ
Программа «Современное программирование»

Что такое Трекер

Все сервисы

Yandex Tracker

Задачи

Проекты

Очереди

Доски задач

Создать задачу

Ещё

Уведомления

Поиск

Документация

Обратная связь

Настройки

Учётная запись

Проекты

Создать проект

Найти проект по названию или описанию

Мои проекты

Внедрение ERP

Черновик

★

Новый буклет про компанию

В работе

☆

Проект внедрения новых методик Аргохолдинга

Черновик

☆

Онбординг

В работе

☆

Промо-кампания

Черновик

☆

Наш новый сайт

Черновик

☆

Внедрить новую методику ведения проектов

В работе

☆

Ежегодная конференция

Черновик

★

Тестирование

Черновик

☆

Полный список моих проектов

Все проекты

Ответственный

Участники

ЕК

Елизавета Колпакова

ИП

ЕК

Елизавета Колпакова

ЕК

Елизавета Колпакова

АВ

Андрей Белов

АВ

Андрей Белов



Зачем Трекеру ограничивать запросы

- Обращение к некоторым подсистемам вычислительно намного дороже, чем к остальным. Их перегрузка может нарушить работу сервера.
- Пользователи иногда пишут ужасные скрипты, обращающиеся к ним по 500 раз.
- Этой же уязвимостью могут воспользоваться злоумышленники

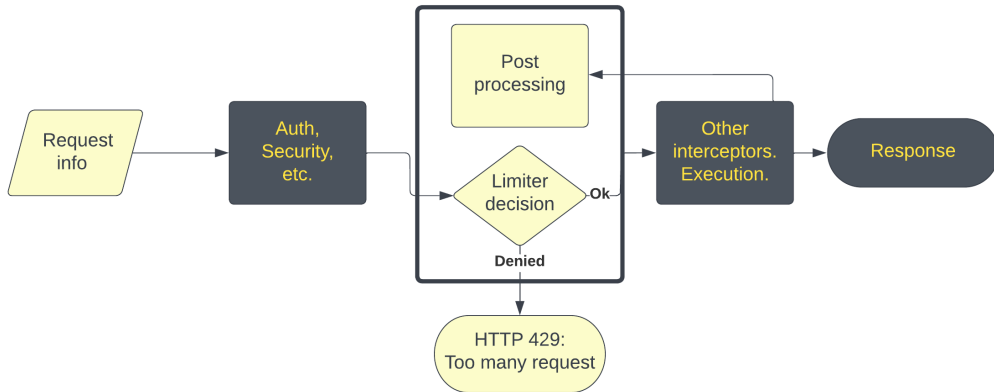


Задачи

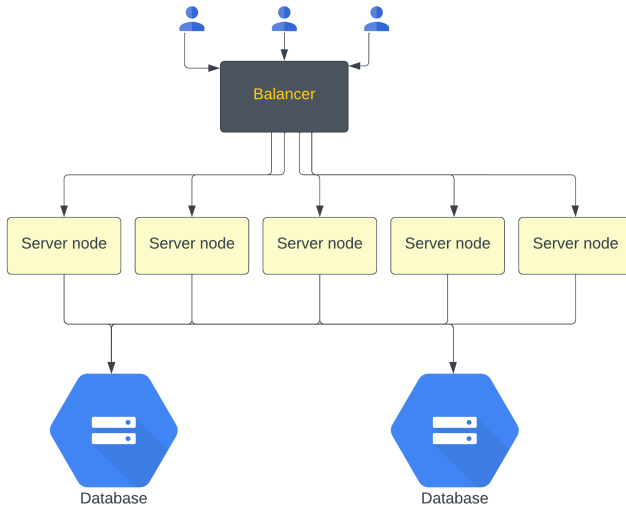
1. Установить проблемы и требования.
2. Исследовать существующие решения.
3. Согласовать и реализовать подход.
4. Протестировать корректность и производительность.



Последовательность обработки запроса



Распределенная структура бэкенда



Требования:

- Быстро принимать решение по запросу, отсутствие сетевых блокировок
- Целостность данных при потере 1 сервера (особый случай обновления)
- Возможность настройки лимитов и параметров без перезапуска

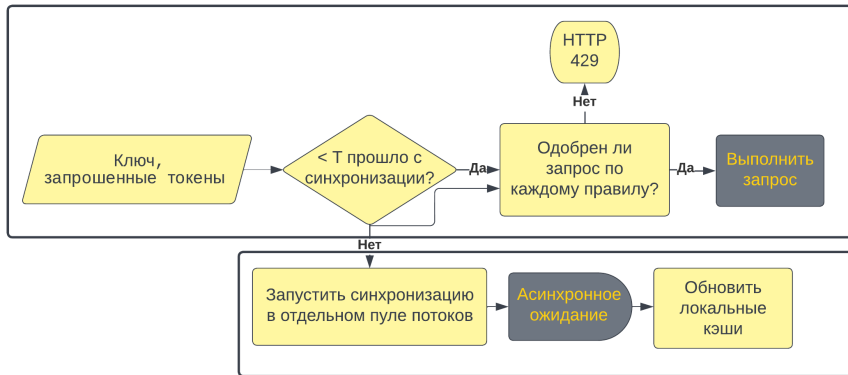


Точность за счёт распределенных хранилищ

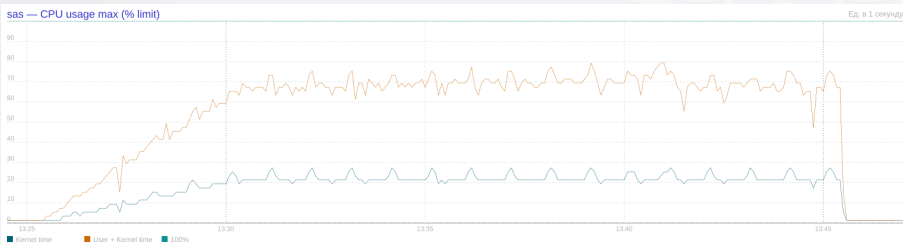
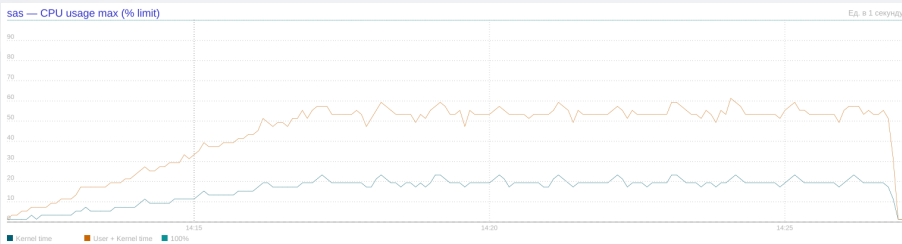
- Предоставляют доступ к общему состоянию (например реализуя распределенный HashMap)
- Множество решений, заявляющих высокую производительность: Redis, Ignite, Hazelcast, Infinispan...
- Для проекта выбран Apache Ignite, оставлена возможность легко перейти на другое решение.



Алгоритм и оптимизация скорости.

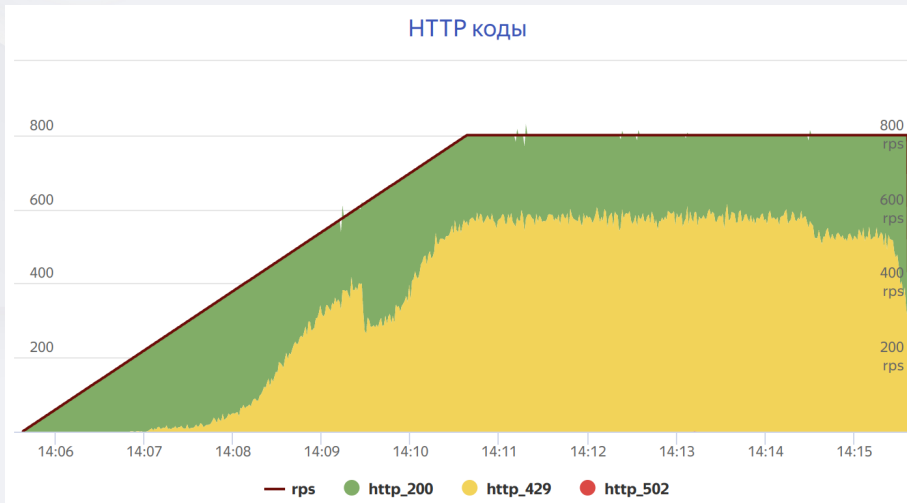


Тесты. CPU





Стресс тесты корректности



Основные трудности

- Распределенное развертывание не быстрое, прототипирование занимало много времени.
- В выбранной библиотеке для лимитирования не было важной для проекта функциональности.
- Не всегда доступны мощности для проведения нагрузочных тестов.



Результаты

1. Разработан и протестирован готовый к использованию в Трекере инструмент.
2. Предоставлен отчёт для команды, содержащий обоснование принятых решений, рекомендации по настройке и работе, а также результаты тестов и их интерпретацию.
3. Измененная копия отчёта в свободном доступе.



Ссылки

Автор: Никита Митцев, muldrik@yandex.ru



Итоговый отчёт в Google Docs



Итоговый отчёт на github

