

Краткий отчет о реализации шардированной системы диалогов

Выполненные работы

1. Архитектурное проектирование

- Декомпозиция предметной области: выделены сущности Message, DialogShard
- Выбор ключа шардирования: композитный ключ на основе GUID обоих пользователей
- Стратегия распределения: детерминированный хеш-алгоритм для равномерного распределения

2. Реализация ключевых компонентов

ShardKeyGenerator

- Создание детерминированных GUID на основе идентификаторов пользователей
- Сортировка ID пользователей для консистентности ключей
- Алгоритм равномерного распределения с защитой от "эффекта Леди Гаги"

ShardManager

- Динамическое управление шардами с взвешенным распределением
- Интерфейс для миграции данных между шардами

DialogService

- CRUD операции с учетом шардирования
- Эффективные запросы к диалогам (всегда в одном шарде)
- Агрегирующие запросы по всем шардам для истории пользователя

3. Решение ключевых требований

Горизонтальное масштабирование

- Ключ шардирования: (user1_id, user2_id) → GUID
- Распределение: равномерное по всем активным шардам
- Вес шардов: поддержка шардов разной мощности

Защита от "эффекта Леди Гаги"

- Композитный ключ распределяет диалоги активного пользователя по разным шардам
- Взвешенное распределение нагрузки

4. Технические особенности

- Типы идентификаторов: все ID - GUID
- База данных: PostgreSQL с оптимизированными индексами
- Docker: контейнеризация с отдельными шардами БД

Преимущества реализации

Производительность

- Локализация данных: весь диалог в одном шарде
- Эффективные индексы: быстрый поиск по пользователям
- Параллельные запросы: для агрегирующих операций

Отказоустойчивость

- Изоляция сбоев: проблемы в одном шарде не затрагивают другие
- Мониторинг: логирование операций шардирования

P.S. В данный момент сегмент dialogs выпал из системы авторизации и не требует token'ов, будет поправлено в последующих итерациях