# Отчет по домашнему заданию № 5 – Масштабируема подсистема диалогов. (Шардирование – практическое применение)

В результате выполнения ДЗ был создан базовый скелет микросервиса, который позволяет записывать диалоги между пользователями с использованием шардирования диалогов. Для обеспечения работы приложения SocialNet использовался docker-compose файл из занятия с небольшими модификациями.

В приложении SocialNet были добавлены методы:

1. Post /dialog/send, который посылает сообщение в диалог и в котором передаются параметры:
   * from\_user – id пользователя, посылающего сообщение
   * to\_user – id пользователя – получателя сообщения
   * dialog\_text – текст сообщения
2. Get /dialog/list/{id\_user} – получение списка сообщения данного пользователя, который идентифицируется id\_user

Для сохранения диалогов была создана шардированная таблица dialogs:

CREATE TABLE IF NOT EXISTS public.dialogs

(

from\_user bigint NOT NULL,

to\_user bigint NOT NULL,

dtext character varying COLLATE pg\_catalog."default",

dist\_key text COLLATE pg\_catalog."default" NOT NULL,

created\_at timestamp with time zone NOT NULL DEFAULT now(),

CONSTRAINT fk\_from\_user\_user\_id FOREIGN KEY (from\_user)

REFERENCES public.users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT fk\_to\_user\_user\_id FOREIGN KEY (to\_user)

REFERENCES public.users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

С внешними ключами, ссылающимися на таблицу users (from\_user, to\_user) и специальной колонкой dist\_key, которая используется как ключ шардирования. created\_at – колонка фиксирующая timestamp записи диалога в таблицу.

Ключ шардирования вычисляется как хэш функция от двух значений – from\_user и to\_user, что дает примерно равномерное распределение диалогов по шардам, при этом диалоги между двумя пользователями с учетом отправителя-получателя будут находиться в одном шарде. Также здесь нивелируется эффект Леди Гаги, как для отправителя, так и для получателя, т.к. выбор ключа шардирования позволяет распределять диалоги по шардам равномерно, даже если кто-то очень много всем пишет или наоборот, если кому-то очень много пишут.

Для обеспечения решардинга также на таблице dialogs создается уникальный индекс (пояснения ниже):

CREATE UNIQUE INDEX IF NOT EXISTS key\_created\_unique

ON public.dialogs USING btree

(created\_at ASC NULLS LAST, dist\_key COLLATE pg\_catalog."default" ASC NULLS LAST)

TABLESPACE pg\_default;

который связывает ключ шардинга со временем записи.

Процесс решардинга производится по порядку:

1. docker-compose -p citus up --scale worker=6 -d - создаются 6 экземпляров шардов
2. Создается таблица dialogs и из нее получаем шардированную таблицу с помощью запросов:
   * SELECT create\_reference\_table('users') – для связки по внешним ключам с шардированной таблицы с таблицей users
   * SELECT create\_distributed\_table('dialogs', 'dist\_key');
3. При запущенном приложении SocialNet с помощью Postman посылаем запросы в приложение для создания диалогов. Перед этим нужно заполнить таблицу users пользователями с помощью sql скрипта only\_100\_users.sql. POST запросы в Postman посылаются по адресу http://localhost:8070/dialog/send с json содержимым {"from\_user": "6", "to\_user": "5", "dialog\_text": "Тогда давай, я за!"}
4. Можно послать несколько запросов для формирования диалога между пользователями.
5. docker-compose -p citus up --scale worker=7 -d – добавляем еще один шард
6. Проверяем, видит ли координатор новый шард - SELECT master\_get\_active\_worker\_nodes();
7. Для перебалансировки необходимо установить на узлах wal\_level = logical, для этого делаем следующие команды:
   * alter system set wal\_level = logical;
   * SELECT run\_command\_on\_workers('alter system set wal\_level = logical');
8. Перезапускаем шарды в кластере и перепроверяем wal\_level:
   * docker exec -it citus-worker-7 psql -U postgres
   * show wal\_level;
9. Теперь можно перезапустить перебалансировку на таблице dialogs - SELECT rebalance\_table\_shards('dialogs');
10. Небольшое пояснение – без уникального индекса перебалансировка не получится и здесь можно применить несколько способов для решения проблемы из мануала citus - <https://docs.citusdata.com/en/v7.2/admin_guide/cluster_management.html>. Я использовал решение № 2 - add replica identity from index и создал индекс с помощью запросов:
    * CREATE UNIQUE INDEX key\_created\_unique ON dialogs (created\_at, dist\_key);
    * ALTER TABLE dialogs REPLICA IDENTITY USING INDEX key\_created\_unique;
11. Перебалансировка таблицы dialogs произошла успешно и можно проверить распеределение по шардам - SELECT nodename, count(\*) FROM citus\_shards GROUP BY nodename;

