Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА №2**

по дисциплине

‘Распределённые системы хранения данных’

Вариант №282550

*Выполнил:*

Студент группы P33302

Тюрин Святослав Вячеславович

*Преподаватель:*

Шешуков Дмитрий Михайлович



Санкт-Петербург, 2024

**Задание:**

Цель работы - на выделенном узле создать и сконфигурировать новый кластер БД Postgres, саму БД, табличные пространства и новую роль, а также произвести наполнение базы в соответствии с заданием. Отчёт по работе должен содержать все команды по настройке, скрипты, а также измененные строки конфигурационных файлов.

Способ подключения к узлу из сети Интернет через helios:  
ssh -J sXXXXXX@helios.cs.ifmo.ru:2222 postgresY@pgZZZ  
Способ подключения к узлу из сети факультета:  
ssh postgresY@pgZZZ  
Номер выделенного узла pgZZZ, а также логин и пароль для подключения Вам выдаст преподаватель.

### **Этап 1. Инициализация кластера БД**

* Директория кластера: $HOME/ivo23
* Кодировка: ISO\_8859\_5
* Локаль: русская
* Параметры инициализации задать через аргументы команды

### **Этап 2. Конфигурация и запуск сервера БД**

* Способы подключения: 1) Unix-domain сокет в режиме peer; 2) сокет TCP/IP, принимать подключения к любому IP-адресу узла
* Номер порта: 9550
* Способ аутентификации TCP/IP клиентов: по имени пользователя
* Остальные способы подключений запретить.
* Настроить следующие параметры сервера БД:
  + max\_connections
  + shared\_buffers
  + temp\_buffers
  + work\_mem
  + checkpoint\_timeout
  + effective\_cache\_size
  + fsync
  + commit\_delay

Параметры должны быть подобраны в соответствии со сценарием OLTP:  
50 одновременных пользователей, 4 сессий на каждого; каждая сессия инициирует до 8 транзакций на запись размером 4КБ; обеспечить максимальную производительность.

* Директория WAL файлов: $PGDATA/pg\_wal
* Формат лог-файлов: .csv
* Уровень сообщений лога: NOTICE
* Дополнительно логировать: завершение сессий и продолжительность выполнения команд

### **Этап 3. Дополнительные табличные пространства и наполнение базы**

* Создать новые табличные пространства для различных таблиц: $HOME/kln97, $HOME/ova89, $HOME/tlv4
* На основе template0 создать новую базу: sickredmath
* Создать новую роль, предоставить необходимые права, разрешить подключение к базе.
* От имени новой роли (не администратора) произвести наполнение ВСЕХ созданных баз тестовыми наборами данных. ВСЕ табличные пространства должны использоваться по назначению.
* Вывести список всех табличных пространств кластера и содержащиеся в них объекты.

**Выполнение:**

Подключение к узлу:

ssh -J s282509@helios.cs.ifmo.ru:2222 postgres0@pg134

TLW6n4pR

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Инициализация кластера БД:

1. Создание директории кластера БД – mkdir -p $HOME/ivo23
2. Изменение системного пользователя - chown postgres0 $HOME/ivo23
3. Инициализация кластера - initdb --encoding=ISO\_8859\_5 --locale=ru\_RU.ISO8859-5 --username=postgres0 -D $HOME/ivo23

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Конфигурация и запуск сервера БД:

Передача конфигурационных файлов через scp:

Передаем на гелиос конфигурационные файлы:

scp postgres0@pg134:ivo23/pg\_hba.conf .

scp postgres0@pg134:ivo23/postgresql.conf .

scp [postgres0@pg134:ivo23/pg\_ident.conf](mailto:postgres0@pg134:ivo23/pg_ident.conf) .

Обратно отправляем на узел:

scp pg\_hba.conf [postgres0@pg134:ivo23/pg\_hba.conf](mailto:postgres0@pg134:ivo23/pg_hba.conf)

scp postgresql.conf [postgres0@pg134:ivo23/postgresql.conf](mailto:postgres0@pg134:ivo23/postgresql.conf)

scp pg\_ident.conf [postgres0@pg134:ivo23/pg\_ident.conf](mailto:postgres0@pg134:ivo23/pg_ident.conf)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**pg\_hba.conf**

Разрешаем подключение Unix-domain сокет в режиме peer и сокет TCP/IP с аутентификацией по имени пользователя, остальные способы подключения запрещаем:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**postgresql.conf**

Меняем следующие параметры:

порт и порты слушания:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

ставим максимальное количество подключений равным 200(50\*4):



Установка размеров буферов shared\_buffers, temp\_buffers и work\_mem:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

shared\_buffers: параметр, который задает количество памяти, которое PostgreSQL будет использовать для кэширования данных из таблиц и индексов в оперативной памяти.

Из документации PostgreSQL следует, что shared\_buffers следует держать от 25% до 40% от всего выделенного ОЗУ. Так как в варианте не сказано сколько ОЗУ выделено на кластер, то будем использовать наиболее частое значение 8Гб, тогда на shared\_buffers выделим 2Гб.

temp\_buffers: параметр устанавливает максимальный размер оперативной памяти, которую сервер может использовать для хранения временных файлов сессии, создаваемых в процессе выполнения операций сортировки и объединения данных.

Размер временных буферов не должен быть большим так как для каждой сессии необходимо обработать 8 транзакций с размером в 4кБ, установим минимум.

work\_mem: параметр отвечает за количество памяти, выделяемой для выполнения операций сортировки, хэширования, агрегирования и других операций обработки данных для каждой сессии.

Так как операции маленькие, то значение work\_mem я выставил в 64кБ (объём памяти для внутренних операций сортировок и хэш-таблиц).

- effective\_cache\_size = 4GB. Оставил по умолчанию (должен быть не меньше чем shared\_buffers).

- fsync = on. Оставил данный параметр включенным, чтобы запись на диск происходила. fsync: Параметр определяет, включена ли синхронизация записи на диск в PostgreSQL. Флаг fsync имеет смысл отключать на read-only копиях бд, в других случаях нужно включать для повышения отказоустойчивости независимо от конфигурации системы.

- commit\_delay = 0(мс). Сохранение на WAL начинается сразу после выполнения операции. Изменение задержки перед сохранением WAL имеет смысл только в том случае, если есть возможность протестировать его влияние на общую производительность.

- checkpoint\_timeout = 10min. Параметр checkpoint\_timeout в PostgreSQL определяет интервал времени в секундах между запусками процесса контрольной точки (checkpoint). Поставил значение на 30 секунд, так как предполагаю, что операции частые.

Включаем архивирование и указываем директорию, в которую будут копироваться WAL-файлы:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Формат лог-файлов: .csv

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Уровень сообщений лога: NOTICE



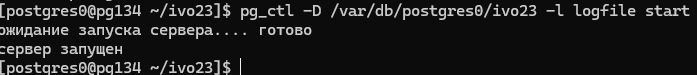
Дополнительно логировать: завершение сессий и продолжительность выполнения команд



### **Этап 3. Дополнительные табличные пространства и наполнение базы**

Запускаем сервер

pg\_ctl -D /var/db/postgres0/ivo23 -l logfile start



Для запуска БД необходимо сначала установить `trust` in `pg\_hba.conf`. Затем можем подключаться.

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Создаем табличное пространство - CREATE TABLESPACE kln97 LOCATION '/var/db/postgres0/kln97';

Создаем табличное пространство - CREATE TABLESPACE ova89 LOCATION '/var/db/postgres0/ova89';

Создаем табличное пространство - CREATE TABLESPACE tlv4 LOCATION '/var/db/postgres0/tlv4';

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

На основе template0 создать новую базу: sickredmath

create database sickredmath with template = template0;



Переподключаемся к базе sickredmath:

psql -h localhost -p 9500 -U postgres0 sickredmath

Создать новую роль, создать таблицы, предоставить необходимые права для заполнения

create role ivan login password 'ivan';

Изображение выглядит как текст, меню, снимок экрана, Шрифт

Автоматически созданное описание

psql -h localhost -p 9500 -U ivan sickredmath -f insert.sql

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Вывести список всех табличных пространств кластера и содержащиеся в них объекты.

select \* from pg\_tablespace;

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

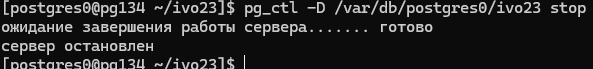
SELECT c.relname, t.spcname FROM pg\_class c JOIN pg\_tablespace t ON c.reltablespace = t.oid;

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Завершаем работу на узле, чтобы не потреблять ресурсы зря.

pg\_ctl -D /var/db/postgres0/ivo23 stop



**Выводы**

Во время выполнения лабораторной работы я научился создавать и конфигурировать кластер БД PostgreSQL. Я познакомился с созданием и работой табличных пространств и ролей.