Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

Рязанский государственный радиотехнический университет

имени В.Ф. Уткина

Кафедра ВПМ

«Распределенные системы обработки информации»

Отчет по лабораторной работе № 4

Выполнил:

студент группы 9413

Чистяков П.А

Проверил:

Князьков П.А

Рязань, 2023

**Цель работы**

Получение практических навыков создания аналитического приложения работы веб-сервиса.

**Условия выполнения**

Файл статистики log.db БД SQLite, полученный по результатам выполнения ЛР 3.

**Задание**

1. Создать веб приложение на основе библиотеки streamlit.
2. Боковая панель приложения должна позволять задавать следующие параметры:

* интервал анализа данных БД в днях (элементы date\_input с проверкой на минимальный интервал в один день);
* интервал разбиения/группировки данных гистограммы в часах (slider с параметрами 1, 24, 4) для п. 4;
* тип операции (text\_input).

1. Из БД необходимо выбрать данные на заданный интервал. На основе выбранных данных необходимо построить гистограмму по количеству вызовов для типа операции (горизонтальной ось гистограммы – типы операций, вертикальная ось – количество вызовов).
2. Из БД необходимо выбрать данные на заданный интервал. На основе выбранных данных необходимо построить гистограмму по количеству вызова для типов операций на сутки (горизонтальная ось гистограммы – время 00:00-24:00 с заданными интервалом «slider» группировки, вертикальная ось – количество вызовов). Если тип операции «text\_input» задан, то гистограмма строится для заданного типа операции, иначе – для всех.
3. Из БД необходимо выбрать данные на заданный интервал. На основе выбранных данных необходимо построить круговую диаграмму по количеству вызовов типа операции (размер сектора типа пропорционален количеству вызовов типа операции).
4. Из БД необходимо выбрать данные на заданный интервал. На основе выбранных данных необходимо построить круговую диаграмму по времени вызовов типа операции (размер сектора типа пропорционален времени выполнения типа операции).

Оформить отчет по результатам выполнения лабораторной работы.

**Ход выполнения работы**

Был разработан клиент для визуализации данных работы сервера. Код клиента представлен на листинге 1. Разработанные формы представлены на рисунках 1-7.

Листинг 1 – Исходный код прокси сервиса

import streamlit as st

import pandas as pd

import datetime

#from sklearn import datasets

import altair as alt

import xmlrpc.client as client

import plotly.express as px

server = client.ServerProxy("http://localhost:8012/RPC2")

st.write("""

# Аналитика работы веб-сервиса

""")

st.sidebar.title("Фильтры")

st.sidebar.header('Дата', divider=True)

start\_date = datetime.date.today()# - datetime.timedelta(days=1)

start\_date = st.sidebar.date\_input(label="С", value=start\_date)

tmp = start\_date + datetime.timedelta(days=1)

end\_date = st.sidebar.date\_input(label="До", min\_value=tmp, value=tmp)

st.sidebar.header('Фильтры')

interval\_width = st.sidebar.slider('Интервал анализа, час', 1, 24, 4)

proc\_type = st.sidebar.text\_input("Тип операции")

with st.spinner('Загрузка...'):

    proc\_type = [] if proc\_type == "" else [proc\_type]

    print(proc\_type)

    data = pd.read\_json(server.slice\_log(proc\_type, start\_date.strftime("%Y-%m-%d %H:%M:%S"), end\_date.strftime("%Y-%m-%d %H:%M:%S"), 0.0, 30000.0))

    fig1 = px.histogram(data, x='action\_name', width=800, height=400)

    fig1.update\_layout(

        title\_text="Гистограмма",

        xaxis\_title\_text="Тип операции",

        yaxis\_title\_text="Количество",

        bargap=0.02,  # Промежуток между столбцами

        bargroupgap=0.1,  # Промежуток между группами столбцов

    )

    fig2 = px.histogram(data, x='date', nbins=int(24/interval\_width), width=800, height=400)

    fig2.update\_layout(

        title\_text="Гистограмма",

        xaxis\_title\_text="Даты",

        yaxis\_title\_text="Количество",

        bargap=0.02,  # Промежуток между столбцами

        bargroupgap=0.1,  # Промежуток между группами столбцов

    )

    # Создаем круговую диаграмму с использованием plotly

    fig3 = px.pie(data, names='action\_name', title='Соотношение количества зопросов по типу')

    fig4 = px.pie(data, names='action\_name', values="duration", title='Соотношение времени запросов по типу')

    st.plotly\_chart(fig1)

    st.plotly\_chart(fig2)

    st.plotly\_chart(fig3)

    st.plotly\_chart(fig4)

    st.subheader('Все данные лог-файлов')

    st.write(pd.read\_json(server.slice\_log()))

    st.subheader('Отобранные данные лог-файлов')

    st.write(data)

st.success('Done!')

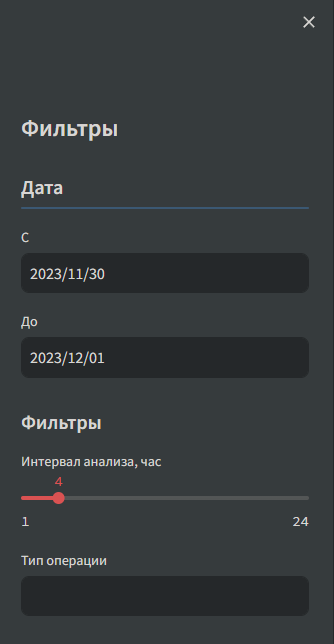


Рисунок – Параметры фильтрации



Рисунок – Отобранные данные

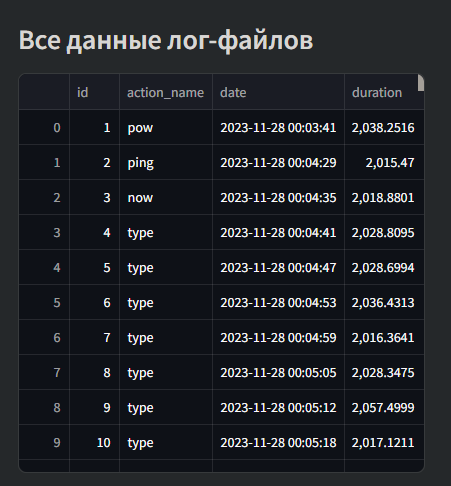


Рисунок – Полный набор данных

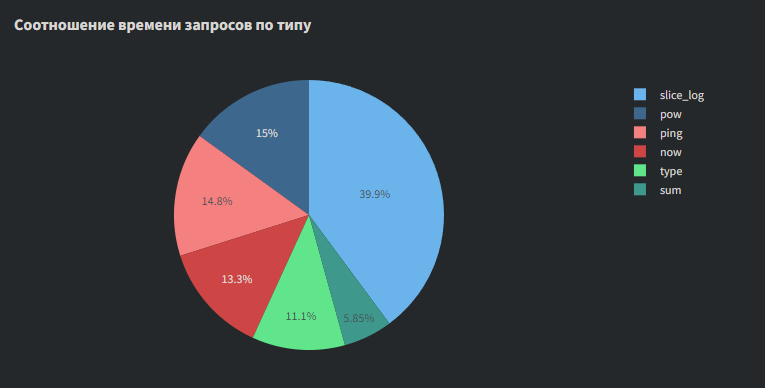


Рисунок – Соотношение времени запросов по типу

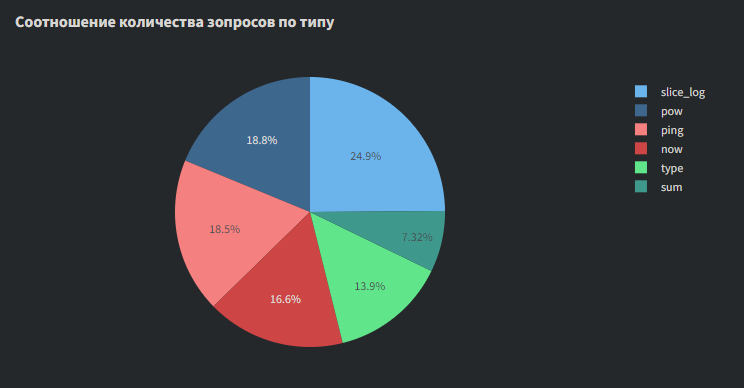


Рисунок – Соотношение количества запросов по типу

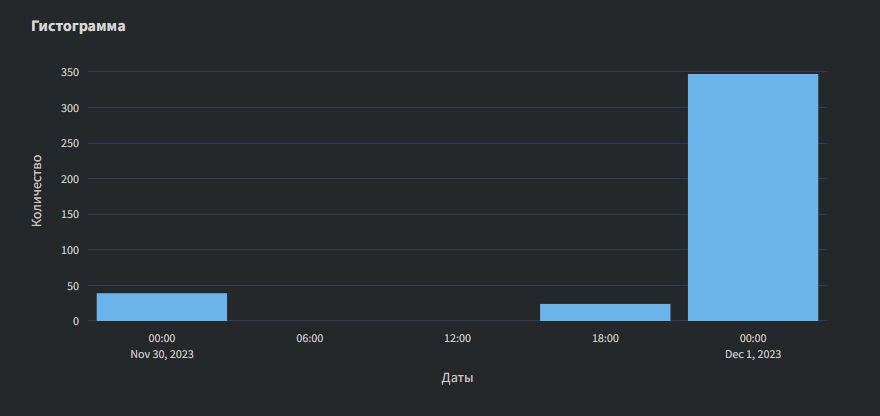


Рисунок – Кол-во запросов в периоды времени

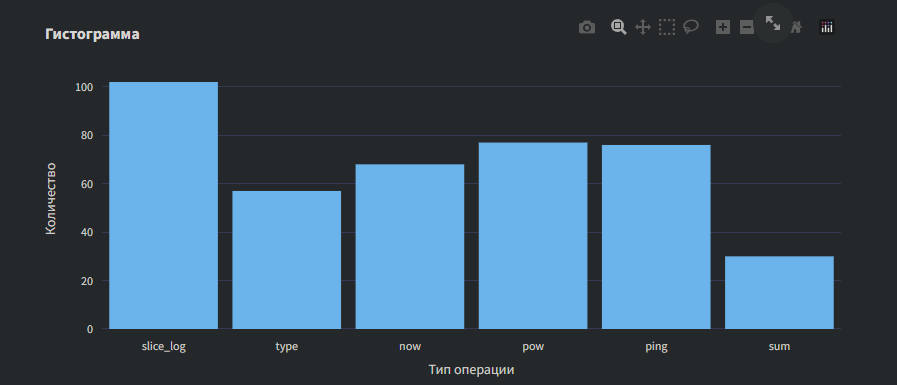


Рисунок – Кол-во запросов по типу

Результаты работы, разработанных серверов прикладываются к данному отчету отдельно.

Проект на github: https://github.com/paavill/distributed-information-development-systems

**Вывод**

В результате выполнения лабораторной работы были получены навыки работы с streamlit Библиотекой.