

SPD eventindex

Разработка каталога физических событий эксперимента SPD на ускорителе частиц NICA: создание программного и пользовательского интерфейса с использованием RestFul технологий

Гурциев Ричард

студент 4 курса бакалавриата, Северо-Осетинский Государственный университет им. Коста Левановича Хетагурова

Научные руководители:

Прокошин Федор Валерьевич(ОИЯИ), Тваури Инга Васильевна(СОГУ)

Задача

Разработать глобальный каталог используя уже имеющийся опыт в эксперименте Atlas, с дальнейшим использованием в установке SPD на коллайдере NICA.

Ожидаемая частота событий в год составит:

30млрд



WEB (MM)



Основные данные

Поля в таблице events

```
run_number | event_number | olf_result | fuid_raw | fuid_aod [|  
    fuid_aod_v2 [| fuid_aod_v3 ...]] | var_1 [var_2] ...
```

olf_result - результат онлайн фильтра, критерии по которым отобрано событие

fuid_raw, ... - UUID для файла в котором хранится это событие, в файле будет храниться от тысячи до 10 тысяч событий, по uuid к файлу можно получить доступ посредством распределенной системы хранения

Поля в таблице datasets

```
fuid_raw | dsid_raw [ fuid_aod | dsid_aod] | date | events | ...
```

dsid_raw, ... - id датасета, в который объединены файлы с raw-данными

events - число событий в датасете

PostgreSQL - это СУБД, которая поддерживает широкий спектр функциональных возможностей, включая сложные запросы, представления и многопользовательский доступ.



connect to db

```
1 from config import DB_HOST, DB_NAME, DB_PASS, DB_PORT, DB_USER  
2  
3 DATABASE_URL = f'postgresql+psycopg://{DB_USER}:{DB_PASS}@{DB_HOST}:{DB_PORT}/{DB_NAME}'  
4 Base = declarative_base()  
5 metadata = MetaData()  
6  
7 engine = create_async_engine(DATABASE_URL)  
8 async_session_maker = sessionmaker(engine, class_async_session, expire_on_commit=False)  
9  
10 async def get_async_session() -> AsyncGenerator[AsyncSession, None]:  
11     async with async_session_maker() as session:  
12         yield session
```



FrontEnd (примеры)



Angular - платформа, которая содержит в себе компоненты web-разметки и функциональной части для создания веб-приложения.

SEARCH ALL

Enter run_number type

Enter event_number version

GET EVENT

Choose File No file chosen UPLOAD FILE

select date

select dataset

select percent

GET EVENT

run_number	event_number	type	version
280308123	129	RAW	t0001
280308123	127	RAW	t0001

status	created_at	file_name
pending	Fri Mar 22 2024 17:48:25	short.txt
success	Fri Mar 22 2024 16:15:28	test.txt
success	Fri Mar 22 2024 16:12:57	testtt.txt
success	Fri Mar 22 2024 14:11:26	push.txt

Add Admins (+)

username	email	active	role	created_at	photo	action
r1char9	vilovnok@gmail.com	True	admin	Sun Mar 24 2024 23:31:13		Edit Delete
flutmyappgg	lutmyappgg@gmail.com	True	admin	Sun Mar 17 2024 18:30:31		Edit Delete
tima	aqery3mo@gmail.com	True	user	Sun Mar 24 2024 23:30:17		Edit Delete
uberkinder	jdaniiss3mo@gmail.com	True	user	Sun Mar 24 2024 23:25:01		Edit Delete

BackEnd (примеры)



FastApi - веб-фреймворк для создания высокопроизводительного API, которое за счет асинхронности обрабатывает большое количество запросов.

Fm - класс, который включает методы для обработки файла.

FileManager

```
1 register = kwargs.get('register')
2 details, get_chain_details, update_location,
3 prompt_prompt_input,
4 response_to_response,
5 detail_chain_details,
6 tags = kwargs.get('tags')
7 total_tokens_total_tokens,
8 total_cost_total_cost
9 )
```



Auth	
POST	/auth/register Register
POST	/auth/login Login
POST	/auth/refresh-token Refresh Token
GET	/auth/update-pass/(email) Register

Verify	
GET	/verify/get-code/(user_id) Get Verify Code
POST	/verify/send-code Send Code
GET	/verify/verify-user/(user_id) Get Verify User Data

User	
GET	/user/profiles Get Users Profiles
POST	/user/profile Get User Profile
POST	/user/save Save User Profile
POST	/user/add Add User Profile
POST	/user/remove Remove User Profile
GET	/user/get-profile-by-username/(username) Get Profile
GET	/user/get-profile-by-email/(email) Get Profile

Search	
POST	/search/get-fuid-raw Get Full Raw
POST	/search/upload-file Upload File
GET	/search/users Get Users
GET	/search/get-all-events Get All Events
GET	/search/get-data-inflow-log Get Dates
GET	/search/get-count-events/(data) Get Count Events
GET	/search/get-certain-events/(data) Get Certain Events

Pipe	
GET	/pipe/pipes Get Pipes
GET	/pipe/load-file/(task_id) Load File
POST	/pipe/spec-file-by-date Get Spec File By Date
GET	/pipe/spec-file/(filename) Get Spec File

BackEnd (Асинхронные результаты)



Redis - это система управления базами данных, используемая для хранения данных в виде ключей и значений в памяти. Также обеспечивает возможность для кэширования, очередей сообщений и других приложений.



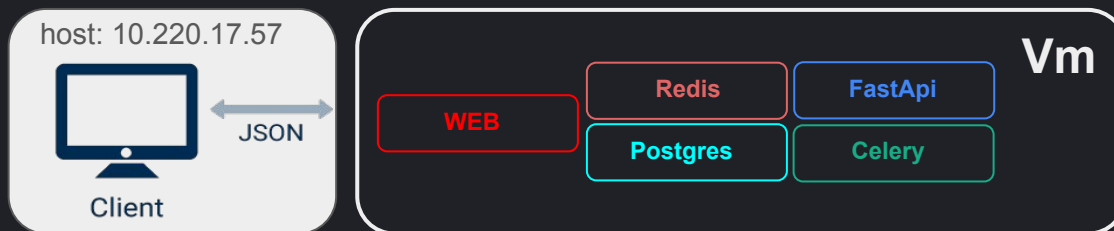
Celery - это система, позволяющая обрабатывать большое количество задач в фоновом режиме, а также распределять их между несколькими воркерами.

В случае если пользователь делает запрос, выполнение которого занимает много времени, результат можно будет увидеть в личном кабинете или на почте. Также в личном кабинете можно будет отслеживать статус выполнения задачи. Система асинхронного выполнения задач работает на основе **Redis** и **Celery**.

devOps



Docker - это платформа для разработки, доставки и запуска приложений в контейнерах. Контейнер представляет собой локальное окружение, в котором будет находиться целая/часть приложения с необходимыми инструментами.



Итог

Текущий результат

- Создан интерфейс, через который пользователи могут взаимодействовать с данными
- Создана серверная часть, которая обрабатывает запросы пользователя
- Прототип системы доступен для тестирования на виртуальных машинах

Дальнейшие задачи

- поработать над стилем сервиса
- интегрировать airflow в систему
- интегрировать kubernetes в систему
- тестирование проекта и добавление функционала

...