

Московский авиационный институт (национальный исследовательский университет)
Институт № 8 «Компьютерные науки и прикладная математика»
Кафедра № 806 «Вычислительная математика и программирование»

**Создание модуля управления временными рядами сигналов для
системы активного мониторинга сложных технических систем**
Выпускная квалификационная работа бакалавра

Студент группы М8О-407Б-19: Инютин Максим Андреевич
Научный руководитель: без степени, без звания,
руководитель центра практик Архитектуры ПАО МТС, Д. В. Дзюба

Москва — 2023



- Актуальность темы данной работы связана с распространением цифровых двойников объектов и систем. Их создание позволяет моделировать отдельные процессы или объекты целиком, проводить тесты, анализировать полученные данные для подбора оптимальных параметров системы.
- Один из способов создания цифрового двойника — установка сенсоров и сбор данных с них. Полученную информацию необходимо систематизировать и хранить.
- На предприятии может быть очень много оборудования, поэтому нужно внедрять эффективный и надёжный модуль хранения данных датчиков.
- Одной из актуальных областей применения цифровых двойников является создание системы мониторинга и предиктивной аналитики работоспособности газотурбинного оборудования электростанций.



Цель и задача работы

Цель — разработать модуль, обеспечивающий управление структурой хранения временных рядов и данными сенсоров для системы мониторинга цифрового двойника промышленных электростанций, использующих газотурбинное оборудование.



Задачи:

- 1 спроектировать модель данных дерева организационной структуры предприятия;
- 2 описать способы взаимодействия: добавление, удаление и изменение вершин и рёбер дерева;
- 3 спроектировать модель хранения временных рядов датчиков;
- 4 изучить средства и технологии, которые будут применяться в ходе разработки программного продукта;
- 5 реализовать модуль управления графом организационной структурой и данными;
- 6 разработать алгоритм объединения данных датчиков с разными частотами дискретизации;
- 7 реализовать генерацию данных для таблиц датчиков, алгоритм получения наборов временных рядов;
- 8 произвести тест производительности реализованного модуля.



Исходные данные:

- Для работы системы мониторинга и предиктивных моделей с объектов предприятия собираются исходные данные.
- Оборудование оснащено датчиками, собирающими данные с разными частотами дискретизации.
- Так как оборудование взаимосвязано и образует сложные технологические цепочки, границы принадлежности датчика к тому или иному оборудованию размыты.

Необходимо разработать модуль, обеспечивающий управление структурой хранения временных рядов и данными сенсоров. Временные ряды должны храниться в ClickHouse, а справочники в PostgreSQL.



Постановка задачи

При реализации необходимо предусмотреть следующие особенности:

- возможность определять структуру таблиц — наборы и типы датчиков к привязке к организационной структуре;
- частота дискретизации датчиков может быть разной: в какой-то момент времени не у всех датчиков есть значение, тогда это событие достраивается по последнему известному значению на этот момент;
- датчики имеют глобальные уникальные идентификаторы;
- для получения данных должна быть возможность получать вектора (все значения датчиков), временной ряд, набор рядов;
- механизм настройки, который будет позволять сопоставлять код датчика к организационной единице;
- дерево оборудования связано с сигналами отношением многие ко многим.

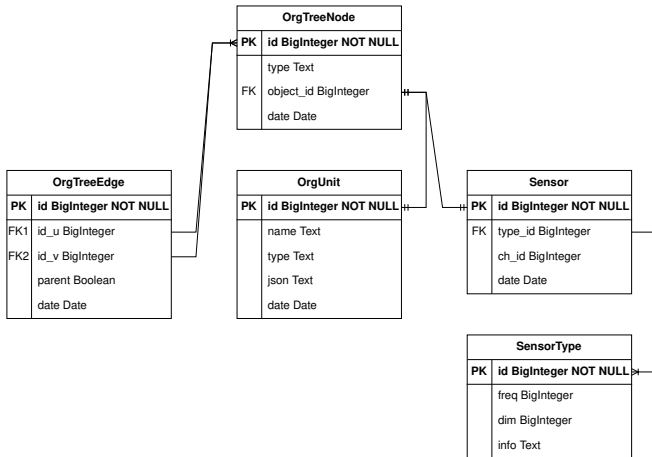


- **Python** является основным языком программирования, который использовался при решении задач;
- **FastAPI** реализует веб-интерфейс для взаимодействия с модулем и базами данных, **SwaggerUI** визуализирует веб-интерфейс;
- **SQLAlchemy** позволяет работать с базами данных на основе объектно-ориентированного подхода;
- **PostgreSQL** обеспечивает хранение дерева организационной структуры предприятия и информации о датчиках;
- **ClickHouse** хранит большие объёмы данных, получаемые от сенсоров;
- **Docker** позволяет разворачивать и переносить изолированные контейнеры с базами данных;
- **GraphViz** визуализирует дерево организационной структуры.



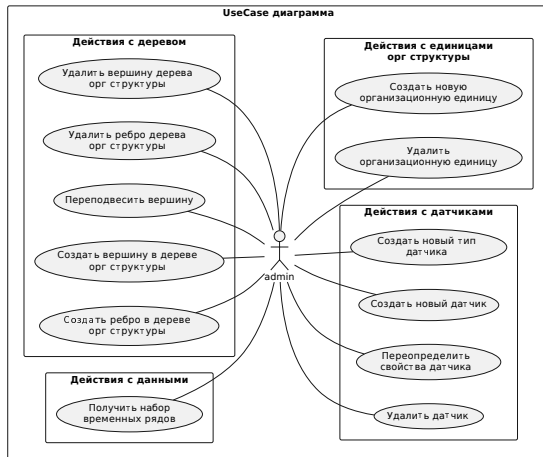
Архитектура решения, алгоритм решения задачи

Модель графа организационный структуры:



Архитектура решения, алгоритм решения задачи

Основные способы управления временными рядами и деревом орг. структуры

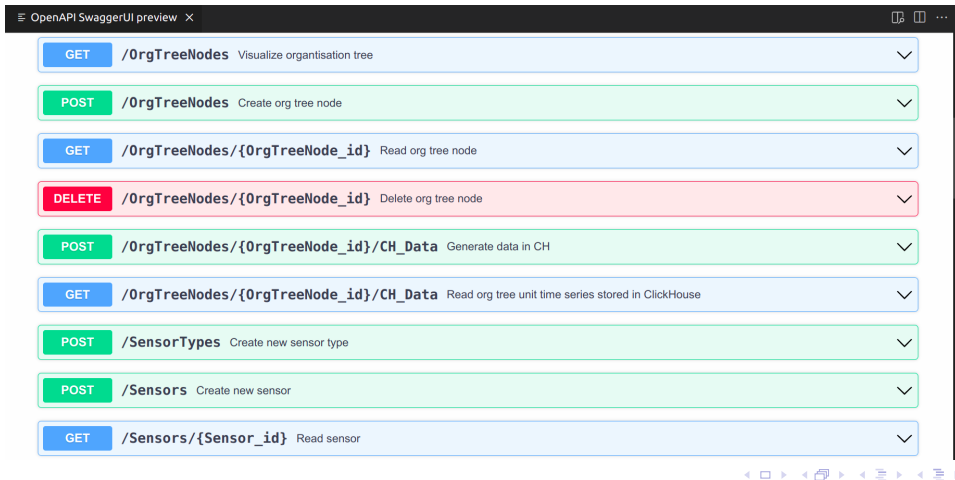


QR-код со ссылкой на GitHub репозиторий с исходным кодом



Результаты разработки

Для демонстрации программного продукта используется OpenAPI SwaggerUI



The screenshot displays the OpenAPI SwaggerUI interface with a list of endpoints. The endpoints are color-coded by HTTP method: GET (blue), POST (green), and DELETE (red). Each entry includes the method, the endpoint path, a brief description, and a dropdown arrow for further details.

Method	Endpoint	Description
GET	/OrgTreeNodes	Visualize organisation tree
POST	/OrgTreeNodes	Create org tree node
GET	/OrgTreeNodes/{OrgTreeNode_id}	Read org tree node
DELETE	/OrgTreeNodes/{OrgTreeNode_id}	Delete org tree node
POST	/OrgTreeNodes/{OrgTreeNode_id}/CH_Data	Generate data in CH
GET	/OrgTreeNodes/{OrgTreeNode_id}/CH_Data	Read org tree unit time series stored in ClickHouse
POST	/SensorTypes	Create new sensor type
POST	/Sensors	Create new sensor
GET	/Sensors/{Sensor_id}	Read sensor



Справочная информация в SwaggerUI

OpenAPI SwaggerUI preview

```
SensorType {
  id*           SensorType_id integer($int64)
                The unique identifier of a sensor type

  freq*        integer($int64)
  dim*         integer($int64)
  info*        string
}
```

```
Sensor {
  id*           Sensor_id integer($int64)
                The unique identifier of a sensor

  type_id*      SensorType_id integer($int64)
                The unique identifier of a sensor type

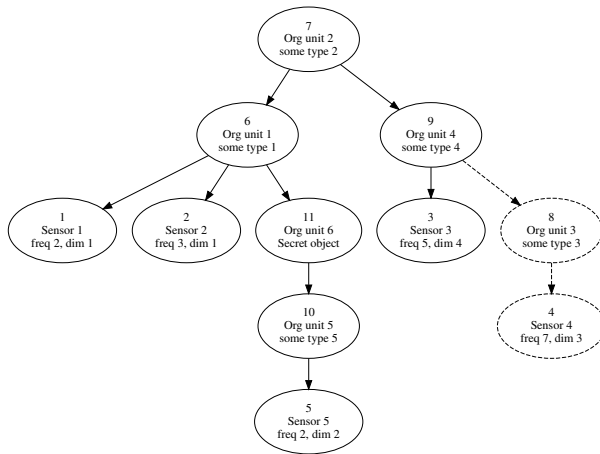
  chid          CH_Sensor_id integer($int64)
                The unique identifier of ClickHouse sensor table

  date*        Date string($date)
                Date of removal
}
```

Error



Визуализация дерева организационной структуры



Результаты разработки

Получение набора временных рядов с датчиков 1, 2, 5

OpenAPI SwaggerUI preview X

Curl

```
curl -X 'GET' \
'http://localhost:8080/OrgTreeNodes/6/CH_Data?time_begin=2023-04-12T14%3A10%3A45.123Z&time_end=2023-04-12T14%3A10%3A55.123Z' \
-H 'accept: text/csv'
```

Request URL

```
http://localhost:8080/OrgTreeNodes/6/CH_Data?time_begin=2023-04-12T14%3A10%3A45.123Z&time_end=2023-04-12T14%3A10%3A55.123Z
```

Server response

Code Details

200

Response body

```
"Time, ms",Sensor_1_value_1,Sensor_2_value_1,Sensor_5_value_1,Sensor_5_value_2
2023-04-12 14:10:44.309000,17290.0,25935.0,34580.0,34580.1
2023-04-12 14:10:44.642333,17290.0,25935.1,34580.0,34580.1
2023-04-12 14:10:44.809000,17290.1,25935.1,34580.2,34580.3
2023-04-12 14:10:44.975667,17290.1,25935.2,34580.2,34580.3
2023-04-12 14:10:45.309000,17290.2,25935.3,34580.4,34580.5
2023-04-12 14:10:45.642333,17290.2,25935.4,34580.4,34580.5
2023-04-12 14:10:45.809000,17290.3,25935.4,34580.6,34580.7
2023-04-12 14:10:45.975667,17290.3,25935.5,34580.6,34580.7
2023-04-12 14:10:46.309000,17290.4,25935.6,34580.8,34580.9
2023-04-12 14:10:46.642333,17290.4,25935.7,34580.8,34580.9
2023-04-12 14:10:46.809000,17290.5,25935.7,34581.0,34581.1
2023-04-12 14:10:46.975667,17290.5,25935.8,34581.0,34581.1
2023-04-12 14:10:47.309000,17290.6,25935.9,34581.2,34581.3
2023-04-12 14:10:47.642333,17290.6,25936.0,34581.2,34581.3
2023-04-12 14:10:47.809000,17290.7,25936.0,34581.4,34581.5
```



Способы хранения временных рядов

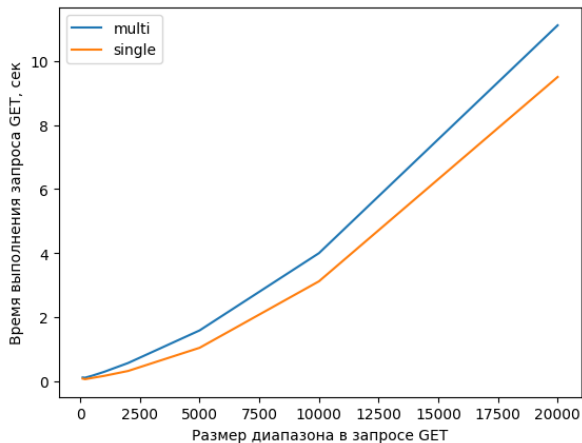
- `multi` — каждому датчику соответствует отдельная таблица в ClickHouse.
- `single` — единая таблица для всех датчиков, в которой предварительно выполнена интерполяция данных.

Необходимо сравнить способы и выбрать более эффективный с точки зрения вычислительных ресурсов.



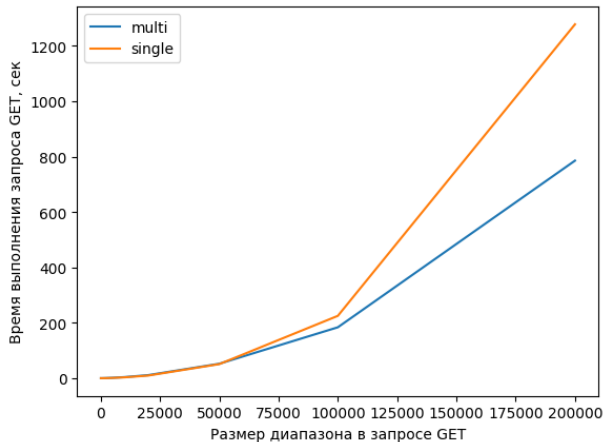
Тесты производительности

Графики времени обработки запроса для диапазонов до 20000 секунд



Тесты производительности

Графики времени обработки запросов для больших диапазонов



- В результате выполнения ВКР был разработан модуль управления временными рядами сигналов сложных технических систем на языке Python с использованием СУБД PostgreSQL и ClickHouse.
- Модуль автоматизирует сбор информации с сенсоров системы, тем самым упрощая создание цифрового двойника электростанции.
- Данные с датчиков надёжно хранятся в базе данных и будут использованы для моделирования объекта и предиктивной аналитики.
- Можно будет оптимизировать работу оборудования, замедляя темпы его износа, повысить отказоустойчивость как отдельной электростанции, так и всей электросети в целом.



Скан отзыва научного руководителя

Скан рецензии

