DOI10.37882/2223–2966.2022.07.14

РАЗРАБОТКА И ВНЕДРЕНИЕ СИСТЕМЫ МОНИТОРИНГА ОПЕРАТИВНЫХ ДАННЫХ ДЛЯ КОНТРОЛЯ ЭЛЕКТРИЧЕСКИХ И ТЕПЛОВЫХ РЕЖИМОВ НА НЕРЮНГРИНСКОЙ ГРЭС

DEVELOPMENT AND IMPLEMENTATION OF A SYSTEM FOR MONITORING OPERATIONAL DATA FOR CONTROL OF ELECTRIC AND THERMAL CONDITIONS AT NERYUNGRINSKAYA GRES

I. Karkokha
V. Samokhina

Summary. This article discusses the process of developing the software and hardware and the user web interface for the monitoring system of operational data (electrical and thermal modes), in order to facilitate the control of the operating personnel.

Keywords: Python, Django, MQTT, Nginx, electrical and thermal parameters, JavaScript.

Каркоха Иван Сергеевич
Технический институт (филиал) ФГАОУ ВО «Северо-Восточный федеральный университет имени М.К. Аммосова» в г. Нерюнгри
tetsik997@gmail.com
Самохина Виктория Михайловна
Доцент, Технический институт (филиал) ФГАОУ ВО «Северо-Восточный федеральный университет имени М.К. Аммосова» в г. Нерюнгри
vsamokhina@bk.ru

Аннотация. В данной статье рассматривается процесс разработки программно-аппаратной части и пользовательского веб-интерфейса для системы мониторинга эксплуатационных данных (электрических и тепловых режимов), с целью облегчения ведения контроля оперативным персоналом.

Ключевые слова: Python, Django, MQTT, Nginx, электрические и тепловые параметры, JavaScript.

На Нерюнгринской ГРЭС применяется множество комплексов для мониторинга, контроля и хранения оперативных тепломеханических и электрических параметров:

- программно-аппаратный комплекс РСДУ-5, обеспечивающий исполнение функций оперативно-технологического контроля и управления в электроэнергетике для генерирующих и распределительных сетевых компаний, крупных промышленных предприятий и предприятий нефтедобывающей и нефтеперерабатывающих отраслей;
- программный комплекс схема потребления тепла (СПТ) программа контроля параметров теплоносителя по очередям теплотрасс г. Нерюнгри. Программа предназначена для диспетчера района тепловых сетей;
- программный комплекс LERS, предназначенный для коммерческого учета тепла.
- Удобство мониторинга, оперативным персоналом, данных комплексов, затруднено следующими причинами:
- множество отвлекающих, не первостепенных параметров;
- в момент переключения между окнами, которые закрывают другие комплексы, оперативный персонал теряет из вида важные параметры;
- при обслуживании оборудования данных комплексов, нет резервирования важных параметров.
- Таким образом, необходимо объединение данных режимов единым интерфейсом и вынесение их на отдельное информационное табло.

Нами были выделены следующие составляющие:

Электрический режим: показания мощности работы турбинных генераторов, передаваемая мощность на линиях электропередач, напряжение, частота сети, суммарный переток.

Температурный режим: температура теплоносителя, расход тепла в час, массовый расход, и давление.

Этапы программной реализации можно разделить на разработку программно-аппаратной части и пользовательского веб-интерфейса.

Исходя из требований разработки, локальный веб-интерфейс быть расположен на веб-сервере, используя следующие технологии:

Nginx — это высокопроизводительный сервер, который реализует функции прокси для веб-серверов и почтовых серверов и потребляет очень мало системных ресурсов.

```
client.on('connect', function () {
client.subscribe([
"CET/1/0/0", /*!топики*/
«CET/1/0/4»,
///
«CET/202/0/4»,
«CET/201/0/10»
]);
});
```

Рис. 1. Подписка на топики

```
client.on('message', function (topic, message) {
sub = topic.split('/')[1];
pid = topic.split('/').pop();
panel = $('.server.' + sub + '[data-id=»' + pid + '»]');
///
if (!isNaN(parseFloat(data)) && (parseFloat(sub) < 20))
panel.find('.value').text((parseFloat(data) * 0.000315).toFixed(2));
else if (!isNaN(parseFloat(data)) && (parseFloat(sub) > 200) &&
(parseFloat(pid)!= 10))
///
});
```

Рис. 2. Преобразование данных

Systemd — подсистема инициализации и управления службами в Linux.

Gunicorn — WSGI-сервер, построенный таким образом, что с ним может взаимодействовать множество различных веб-серверов.

Была произведена установка интерпретатора объектно-ориентированного языка программирования Python 3.9 и прокси веб-сервера Nginx.

Далее, перейдя в каталог www/home/code и использовав команду gitclone, клонировал, с сервиса gillab, «Чистый шаблон», включающий в себя следующие компоненты: конфигsystemd, nginx, gunicorn и bash-скрипт — install.sh, при помощи которого будет развернут веб-сервер.

Запустив bash-скрипт (install.sh) потребуется ввести домен зарегистрированный на DNS-сервере (plazma.asu.ngres), остальная конфигурация выполнится автоматически.

Приведем подробное описание того, какие команды исполняются при выполнении скрипта:

- base_folder=`pwd | xargsbasename` — получение текущей директории;
- base_python_interpreter=»/home/www/python/bin/python3.9» — путь, где лежит интерпретатор питона;
- read-p"Your domain without protocol (or IP): "project_domain — запрашиваетдоменисохраняетвпеременную;
- `$base_python_interpreter-mvenvenv` — создание виртуального окружения;
- sourceenv/bin/activate — запуск вирутального окружения;
- pipinstall-r requirements.txt — установка необходимых пакетов, заранее прописанных в файле requirements;
- sed -i «s~dbms_template_domain~$project_domain~g» nginx/$base_folder.confsrc/config/settings.py — заменяется название домена в конфигурационном файле nginx и Django;

```
<td class="col-xs-2 ">ТГ-1</td>
<td class="col-xs-2 server 1 text-right" data-id="0">
<span class="value">0</span>
</td>
<td class="col-xs-2">МВт</td>
```

Рис. 3. Рендеринг преобразованных данных

Рис. 4. Окно «Электрические параметры»

- sed-i»s~dbms_template_domain~$project_domain~g»nginx/$base_folder.confsrc/config/settings.py — Устанавливает ссылку на конфигурационный файл nginx;
- sudoln-s$project_path/systemd/$base_folder-gunicorn.service/etc/systemd/system/ — Устанавливает ссылку на конфигурационный файл system.

Была проведена проверка работоспособности службы с помощью команды systemctlstatusplazma-gunicorn.

Убедившись, что служба запущена, запустил стартовую страницу пустого проекта Django, вписав домен, который указывался при запуске bash-скрипта, поисковую строку браузера.

На этом разработку «Программно-аппаратной части» можно считать завершенной.

На этапе создания проекта при помощи скрипта автоматического разворачивания веб-сервера, используя модуль pip, был установлен framework Django 4.0.4. Для создания в структуре проекта Django каталога с конфигурационными файлами выполнил команду в запущенном виртуальном окружении «django-adminstartprojectconfig», результатом команды является каталог, в котором располагаются стандартные Django-файлы. Для того чтобы создать приложение, была выполнена терминальная команду «python manage.py startapp app», организующую структуру каталогов приложений Django. Следующим шагом являлось написания Django шаблонов и реализация верстки веб-страниц. Помимо этого, с официального сайта, был установлен, cssфреймворк Bootstrap. Так как в Django статические файлы (css, js, картинки, видео файлы), обычно, помещаются в специальной папке — static, было необходимо ее создать и поместить в нее, скаченные ранее Bootstrap файлы.

На следующем этапе разработки были решены следующие задачи: подключение к серверу очередей, получение и преобразование данных и их рендеринг. Так как база данных не предусматривалась для данной системы, было необходимо организовать подключение к серверу очередей напрямую из приложения. В ходе поисков оптимального варианта для разрешение поставленной на текущем этапе разработки задачи, была обнаружена, и в последствии использована в итоговом варианте системы, библиотека, написанная на JavaScript: MQTT.js. Это клиентская библиотека для протокола MQTT, написанная на JavaScript для браузера. Натекущее время, это наиболее широко используемая библиотека для работы с MQTT брокерами.

Благодаря однопоточной функции JavaScript MQTT.js является полностью асинхронным клиентом MQTT. MQTT.js поддерживает MQTT/TCP, MQTT/TLS, MQTT/WebSocket, а степень поддержки в различных операционных средах следующая:

- Браузер: MQTT через WebSocket.
- Node.js: MQTT, MQTT через WebSocket.

Для реализации подключения использовался метод — mqtt.connect, в котором указано:

- адрес брокера;
- логин;
- пароль;
- интервал между двумя переподключениями.

Этот метод открывает новое соединение MQTT с указанным брокером. Соединение устанавливается асинхронно, и любая функция, зарегистрированная с помощью mqtt.connect(), будет вызываться вне зависимости от того, будет ли попытка успешной или неудачной.

Следующим шагом, для достижения цели, поставленной на текущем этапе разработки, а именно получение из сервера очередей, параметров, было необходимо реализовать подписку на соответствующие топики. На рисунке 1 представлен пример подписки на топики, в которых хранятся электрические параметры.

На этом этапе, можно считать, что параметры из топиков уже получены, и их можно использовать, но перед тем, как приступить к рендерингу, стоит упомянуть, что, некоторые данные (в основном электрические параметры) полученные, в результате подписки на топики, приходят в виде строки, в нечитабельном формате. Поэтому, из полученных строк требуется спарсить значения тех параметров, которые необходимо отобразить, и с помощью математических формул преобразовать верному формату. Преобразование данных представлено на рисунке 2

Преобразованные данные были отрендерены на веб-страницах следующим образом: в класс тега, для создания ячейки таблицы, td указано второе значение из топика, указывающее на номер счетчика, на который была осуществлена подписка, а в data-id передано крайнее значение указывающее на сам электрический или тепловой параметр. На рисунке 3 представлен наглядный пример реализации.

Входе тестированиясистемы не было выявлено критических ошибок. На всех веб-страницах данные отображаются и обновляются в реальном времени, т.е. когда с счетчиков данные поступают на сервер очередей они моментально отобразятся на веб-страницах; все значения четко видны, цветовая схема не ослепляет персонал в ночное время; навигация, как и было заявлено, осуществлятся с помощью отдельного клавиатурного блока (рис. 4).

В настоящее время система мониторинга внедрена в рабочий процессе.

ЛИТЕРАТУРА

1. Кириченко А.В. Динамические сайты на HTML, CSS, Javascript и Bootstrap. Практика, практика и только практика [Текст] / А.В. Кириченко, Е.В. Дубовик.— 2-е изд.— Санкт-Петербург: Наука и Техника, 2018.— 272 с. [Электронный ресурс] // Режим доступа свободный, URL: https://www.labirint.ru/books/653502/
2. Федоров Д.Ю. Программирование на языке высокого уровня Python: учебное пособие для прикладного бакалавриата / Д.Ю. Федоров.— 2-е изд., перераб. и доп.— Москва: Издательство Юрайт, 2019.— 161 с.
3. Шелудько В.М. Основы программирования на языке высокого уровня Python: учебное пособие / В.М. Шелудько.— Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017.— 146 с.

Журнал «Современная наука: актуальные проблемы теории и практики»