

который подбирает функцию  $f$ , исходя из переданной обучающей выборки. После проведения обучения алгоритм готов к работе на рабочих данных.

*Результаты.* Для проверки алгоритма сформируем выборку из датчиков влажности, которые не участвовали в обучении модели. Наличие шума в них известно исследователю, но неизвестно алгоритму. Проведём с ними те же подготовительные операции, которые проводили с обучающей выборкой: найдём вектор разностей для каждого датчика, и выполним поиск класса для каждого вектора с помощью метода `predict`. Результатом работы метода `predict` является массив, каждый элемент которого означает принадлежность вектора к классу, в данном случае 0 – нет шума, 1 – есть шум. Результат классификации на 100 % совпал с известными данными о зашумлённости показаний.

*Выводы.* Разработанная методика позволяет автоматизированным способом, без субъективной оценки оператора определять наличие шума в показаниях датчиков влажности. С помощью методики может проводиться отбраковка датчиков как до их калибровки, так и после. Наиболее целесообразно проводить отбраковку одновременно с операцией калибровки, так как в этом случае количество технологических операций сокращается, и снижается себестоимость изделий. Методика может быть применена при оценке уровня шума других датчиков, но при этом следует учитывать вид функции, описывающей зависимость выходной величины измерительного преобразователя от измеряемой величины.

### Список использованных источников

1. Вьюгин В.В. Математические основы машинного обучения и прогнозирования / В.В. Вьюгин, ВЦНМО, 2013. – 304 с.
2. Pedregosa F. [и др.]. Scikit-learn: Machine Learning in Python // Journal of Machine Learning Research. 2011. (12). С. 2825–2830.

УДК 004.031.2

**В. А. Каюров, В. Ю. Носков**

ФГАОУ ВО «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина», г. Екатеринбург, Россия

## РАЗРАБОТКА КОМПЛЕКСНОЙ СИСТЕМЫ УЧЕТА ОБОРУДОВАНИЯ И СОБЫТИЙ В ГБУ СО «ОПЕРАТОР ЭЛЕКТРОННОГО ПРАВИТЕЛЬСТВА»

### Аннотация

*В статье описан процесс разработки комплексной системы учета оборудования и событий в ГБУ СО «Оператор электронного правительства». Система находится на начальном этапе разработки. Цель работы – создать набор взаимосвязанных систем инвентаризации оборудования, мониторинга и учета событий в ГБУ СО «Оператор электронного правительства».*

*Разрабатываемая система будет содержать постоянные актуальные сведения об имеющемся оборудовании и его функционировании.*

*Назначение информационной системы:*

- 1. Проанализировать текущее состояние оборудования в организации;*
- 2. Своевременное выявление и устранение неполадок в оборудовании;*
- 3. Централизованный доступ до всей информации об оборудовании.*

*Описывается общая архитектурная схема системы. Рассказывается об основных технических решениях, использованных при разработке системы.*

**Ключевые слова:** *мониторинг, устройство, система, оборудование, скрипт.*

#### **Abstract**

*The article describes the process of developing an integrated system for recording equipment and events in the "Operator of Electronic Government". The system is at the initial stage of development. The purpose of the work is to create a set of interconnected systems for inventorying equipment, monitoring and recording events in the State Enterprise "Operator of electronic government".*

*The system will contain permanent up-to-date information about the existing equipment and its operation.*

*Purpose of the information system:*

- 1. Analyze the current state of equipment in the organization;*
- 2. Timely detection and elimination of malfunctions in equipment;*
- 3. Centralized access to all information about the equipment.*

*It tells about the general architectural scheme of the system. It tells about the main technical solutions used in the development of the system.*

**Key words:** *monitoring, device, system, equipment, script.*

Современный мир непосредственно связан с высокими технологиями, они шагают в ногу со всем человечеством и уже вряд ли когда мы сможем отказаться от них. Их перечень на сегодняшний день очень велик начиная от умных наручных браслетов и заканчивая целыми комплексами, состоящие из большого числа элементов. Их вклад в нашу жизнь неоценим, они приносят нам огромную пользу, облегчают труд и выполняют за нас рутинную работу. Уже невозможно представить какую-либо сферу человечества где бы не применялись современные технологии. Однако, несмотря на легкость применения, за такими технологиями скрывается очень большой человеческий труд. Над одним устройством могут работать сотни людей, общими усилиями которых, по крупицам создаются грандиозные проекты. Любая технология, несомненно, должна развиваться и своевременно обновлять свои функциональные предназначения.

В процессе формирования человеческой деятельности существенную роль занимает её многостороннее наблюдение. Суть состоит в сборе нужных данных и последующем их анализе. Постоянное осуществление мониторинга позволяет вовремя обнаружить ошибки, погрешностей и их устранение в самые короткие сроки. Достаточно часто возникают ситуации, когда необходимо знать состояние системы, к которой нет локального доступа. Отсутствие доступа может вызвано территориальной удаленностью системы или физическими ограничениями безопасности, поэтому возникла необходимость создания инструментов удаленного мониторинга.

*Архитектура системы.* Система представляет из себя набор взаимоувязанных между собой подсистем инвентаризации и мониторинга. Каждая из подси-

стем предназначена для решения специфических задач, для которых она разрабатывалась. Вместе эти подсистемы позволяют собрать всю необходимую и подробную информацию об оборудовании. Общая архитектура системы представлена на рисунке 1.

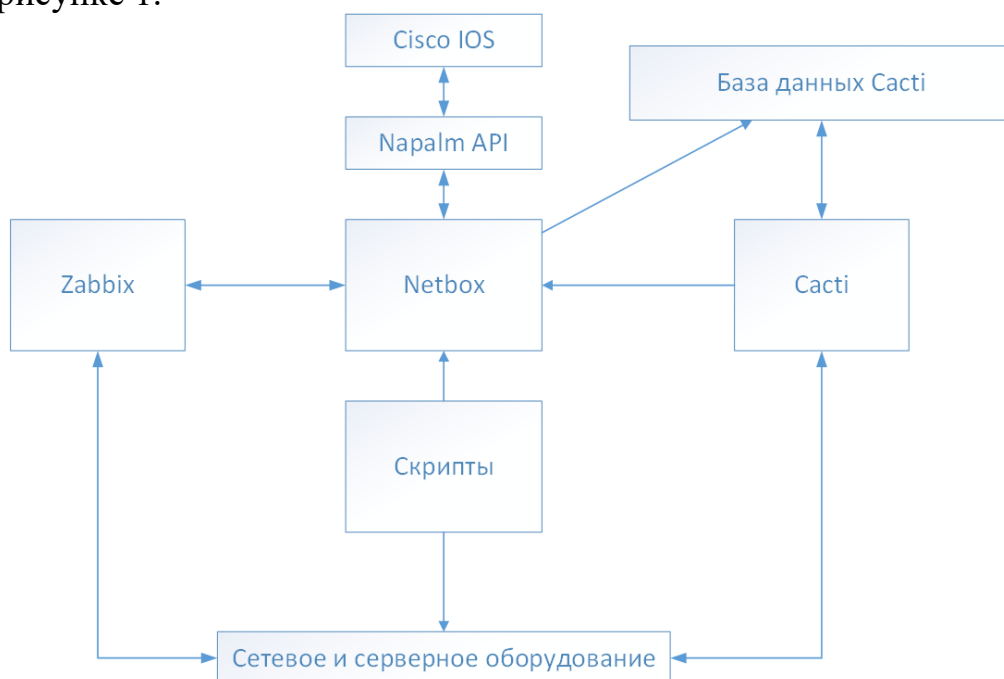


Рис. 1. Общая архитектура системы

За основу разрабатываемой системы было решено взять веб-приложение Netbox. Netbox – это веб – приложение с открытым исходным кодом, предназначенное для управления и документирования компьютерных сетей [1]. Netbox написан на языке Python, что значительно упрощает его модернизацию под собственные нужды.

На сервере, где развернут Netbox, с помощью скриптов можно запрашивать информацию непосредственно с устройств, а также действующих систем мониторинга.

Для сбора необходимой информации с устройств Cisco на сервер была установлена библиотека Python NAPALM, которая реализует набор функций для взаимодействия с различными устройствами поставщика маршрутизаторов с использованием единого API [2].

Для получения графических данных мониторинга будет использована система Cacti [3]. Т.к. у этой системы нет своего API необходимо, работать напрямую с системой не получится. Поэтому для того, чтобы получить данные с нужного устройства, необходимо обращаться к базе данных Cacti. Обращаясь к базе, скрипт получает необходимую информацию, и возвращает прямую ссылку на нужный график.

Система мониторинга Zabbix используется по большей части для получения информации о серверном оборудовании. Его интеграция с Netbox находится в стадии проработки.

Различные скрипты необходимы для получения специфических данных с устройств, либо для получения данных, которые по каким-то причинам не могут предоставить системы мониторинга и API.

Таким образом, предполагается что Netbox выступит в разрабатываемой системе для полной инвентаризации оборудования, а различные действующие системы мониторинга и скрипты будут передавать необходимую информацию в Netbox для того, чтобы вся информация по конкретному устройству находилась централизованно в одном месте.

*Реализация системы.* Разработка информационной системы находится на начальном этапе. Основной язык разработки – Python. Для развертывания информационной системы была выбрана операционная система (ОС) Ubuntu 16.04. Эта ОС поддерживается всеми используемыми открытыми решениями, имеет широкий набор предустановленных библиотек Python. На эту систему были установлены Netbox и NAPALM.

Главная страница Netbox представлена на рисунке 2.

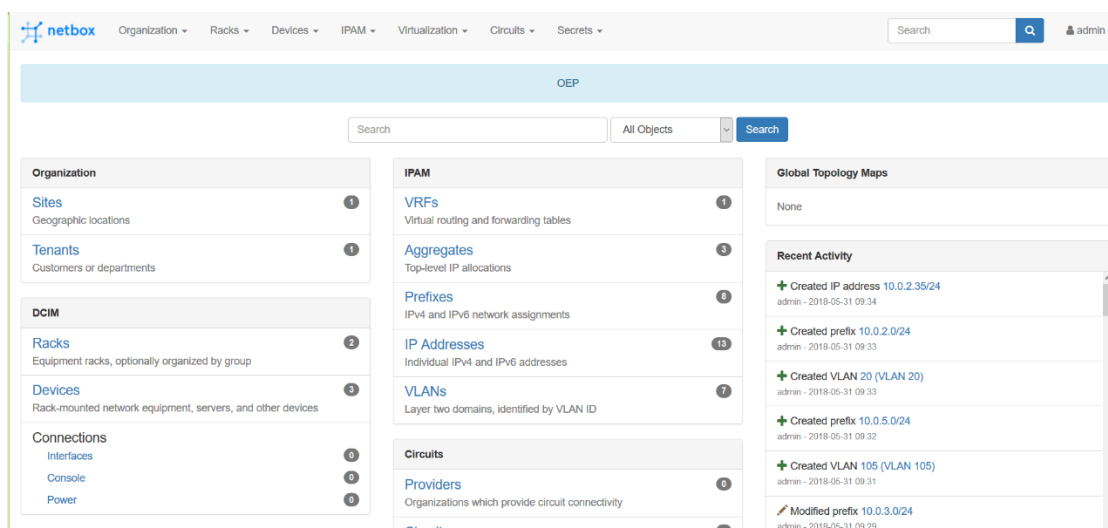


Рис. 2. Главная страница Netbox

Здесь имеется доступ ко всем основным разделам инвентаризации. Для тестирования системы было добавлено 2 сетевых устройства: Cisco 1941 и Cisco 6509E. В Netbox можно указать в какой стойке и в каком юните находится устройство, указать IP адреса, MAC-адреса, соединения и т.д.

Интеграция с NAPALM позволяет получить общую текущую информацию об устройстве. На рисунке 3 представлен скриншот вкладки статус. При выборе вкладки Status с помощью API NAPALM запрашивается информация с устройства на данный момент.

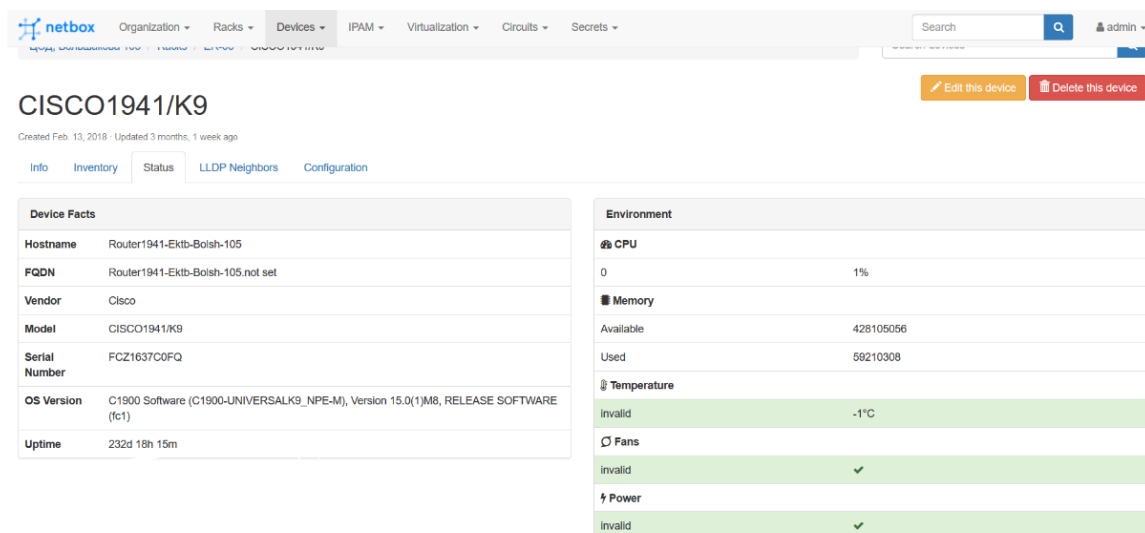


Рис. 3. Скриншот вкладки Status

Аналогичным образом работают вкладки LLDP Neighbors и Configuration. Вкладка LLDP позволяет увидеть соседние устройства на интерфейсах (при условии, что LLDP протокол включен). Вкладка Configuration показывает несколько типов конфигураций устройства: текущую конфигурацию и конфигурацию, которая запускается при запуске устройства.

В Netbox нет функции автоматического добавления интерфейсов и информации о них. С учетом большого количества интерфейсов на некоторых устройствах и для автоматизации процесса добавления интерфейсов был написан скрипт который будет делать это автоматически. Данный скрипт позволяет собрать информацию об интерфейсах с устройств и автоматически добавляет ее в базу данных Netbox. При этом всегда остается возможность вручную исправить или дополнить информацию.

Для отображения графических результатов для каждого интерфейса был добавлен значок графика. Графики будут получены из системы Cacti с помощью скрипта. Суть работы скрипта заключается в получении конечной ссылки на график путем поиска соответствующего устройства и его интерфейса. В дальнейшем этот скрипт можно просто дополнять получением любых графиков, работая с БД Cacti.

Полученные результаты на данном этапе позволяют продолжать разработку системы.

### Список использованных источников

1. Web-ресурс сети Интернет [сайт]. URL: <https://netbox.readthedocs.io/en/latest/>.
2. Web-ресурс сети Интернет [сайт]. URL: <https://na-palm.readthedocs.io/en/latest/>.
3. Web-ресурс сети Интернет [сайт]. URL: <https://www.cacti.net/>.