



WYDZIAŁ
MATEMATYKI
I FIZYKI STOSOWANEJ
POLITECHNIKI RZESZOWSKIEJ

Stefan Domaradzki

Agregacja danych z czujnika i Raspberry Pi przy użyciu
Apache Kafka i języka Python

Praca dyplomowa

Opiekun pracy:

Dr. Marek Bolanowski

Rzeszów, 2023

Spis treści

1. Data streaming	4
2. Raspberry Pi	4
3. Realizacja	4
3.1. Konfiguracja RaspberryPi	4
3.1.1. System Operacyjny	4
3.1.2. Konfiguracja IP	4
3.1.3. DNS resolution	5
3.2. Połączenie	6
3.3. Termometr	7
3.4. Skrypty w języku python	8
3.4.1. Temperature Producer	8
3.5. Consumer.py	8
4. Wnioski	9
5. Pliki projektu	9
Literatura	9

1. Data streaming

Strumieniowanie danych (data streaming) jest procesem przetwarzania danych, który polega na przesyłaniu ich strumieniowo, tzn. w czasie rzeczywistym lub w małych paczkach, zamiast jednorazowo przesłać cały zbiór danych. Strumienie danych pochodzą z różnych źródeł, takich jak sensory lub aplikacje internetowe. Strumieniowanie danych pozwala na przetwarzanie i analizowanie danych w czasie rzeczywistym, co umożliwia szybkie reagowanie na zmieniające się warunki.

2. Raspberry Pi

Raspberry Pi to mały jednopłytkowy komputer o stosunkowo dobrej mocy obliczeniowej. Na jednej małej płytce znajdują się procesor, pamięć RAM, USB, HDMI, czy Ethernet. Architektura płytki została oparta na układzie systemu na chipie (SoC). I jest w stanie uruchamiać standardowe systemy operacyjne, lub dedykowany Raspberry Pi OS.

3. Realizacja

3.1. Konfiguracja RaspberryPi

3.1.1. System Operacyjny

Instalacja systemu operacyjnego na RaspberryPi jest bardzo prosta. Wykorzystane zostaje darmowe oprogramowanie zapewnione przez producenta mikrokomputera o nazwie *Raspberry Pi Imager*". Dzięki niemu mamy dostęp wybrania preferowanego systemu operacyjnego. Wybrano *RaspberryPi OS (32-bit)*. Wybieramy docelowe miejsce instalacji systemu (Karta microSD).

3.1.2. Konfiguracja IP

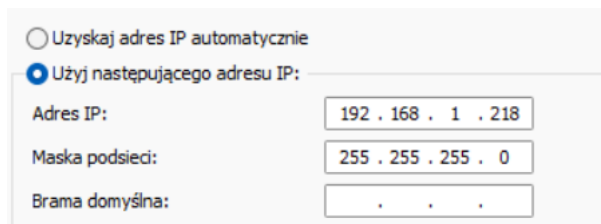
Dla prostszego łączenia się z Raspberry skonfigurowano statyczny adres IP dla połączenia przewodowego. Ponieważ nie istnieje możliwość ustawienia tego parametru z poziomu instalatora *Raspberry Pi Imager*, należy dokonać edycji pliku znajdującego się pod następującą ścieżką `.conf` zgodnie z zamieszczonym rysunkiem.

Oczywiście nadany adres może być dowolny należy jednak pamiętać, że to za jego pomocą następuje połączenie do urządzenia.

W przypadku wiersza *staticrouters* zapisujemy IP urządzenia które służy jako router i jednocześnie Kafka Server.

```
interface eth0
static ip_address=192.168.1.217/24
static routers=192.168.1.218
```

Rysunek 3.1: Konfiguracja pliku dhcpd.conf



The screenshot shows a network configuration window with two radio buttons at the top: 'Uzyskaj adres IP automatycznie' (unselected) and 'Użyj następującego adresu IP:' (selected). Below the selected option, there are three input fields: 'Adres IP:' with the value '192 . 168 . 1 . 218', 'Maska podsieci:' with the value '255 . 255 . 255 . 0', and 'Brama domyślna:' with three empty fields separated by dots.

Rysunek 3.2: Konfiguracja sieci na PC

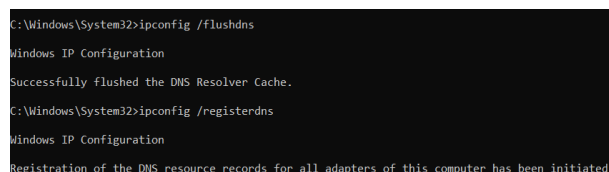
3.1.3. DNS resolution

Jeśli nie jest możliwe przypisanie nazwy hosta do adresu IP, nastąpi błąd w wyszukiwaniu DNS.

Jeśli DNS resolution nie działa poprawnie, należy dodać wpis do pliku */etc/hosts* na Raspberry Pi. Po otwarciu za pomocą dowolnego edytora tekstu należy dodać w nim linijkę z adresem IP brokera Kafka i jego nazwą hosta. Przykładowo:

192.168.1.218 "Nazwa Hosta"

Dobłą praktyką jest również wyczyszczenie pamięci podręcznej i resolverów na maszynie funkcjonującej jako Kafka server Co zagwarantuje aktualność wszystkich rekordów DNS.



The screenshot shows a Windows command prompt with the following commands and output:

C:\Windows\System32>ipconfig /flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.

C:\Windows\System32>ipconfig /registerdns

Windows IP Configuration

Registration of the DNS resource records for all adapters of this computer has been initiated.

Rysunek 3.3: Czyszczenie pamięci podręcznej DNS

3.2. Połączenie

Połączenie następuje poprzez użycie SSH. SSH to standard protokołów komunikacyjnych używanych w sieciach komputerowych TCP/IP. Po podłączeniu Raspberry Pi za pomocą przewodu sieciowego do urządzenia działającego jako serwer Kafki wykonujemy następującą komendę:

SSH pi@192.168.1.217

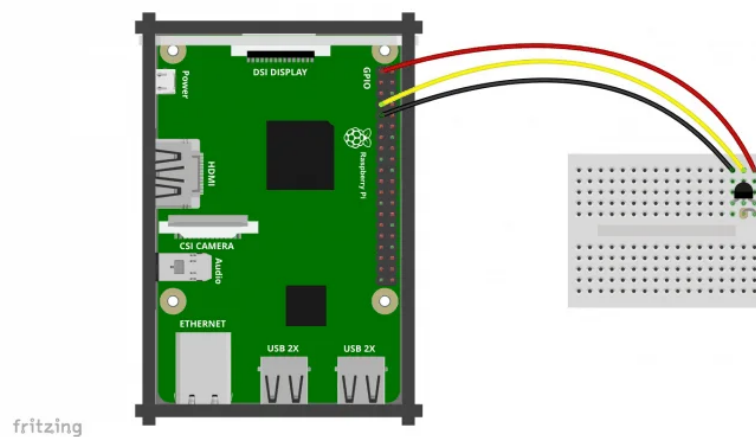
Pamiętając, że jest to wcześniej ustawiony adres używanego mikrokomputera. Sprawdzenie dostępności portu na którym pracuje *Kafka Server*:

```
pi@raspberrypi:~/rozszerzone $ nc -zv 192.168.1.218 9092
Connection to 192.168.1.218 9092 port [tcp/*] succeeded!
pi@raspberrypi:~/rozszerzone $
```

Rysunek 3.4: Test dostępności portu 9092

3.3. Termometr

Wykorzystano następujący schemat połączenia czujnika *DS18B20* z Raspberry Pi:



Rysunek 3.5: Termometr schemat połączenia

Aby jednak można było odczytać temperaturę z czujnika należy włączyć odpowiedni sterownik w jądrze systemu Raspberry.

Aby to zrobić należy włączyć panel konfiguracji:

```
sudo raspi-config
```

I w zakładce "Interfacing Options" włączyć opcję: P7 1-Wire. Na koniec należy zrestartować urządzenie za pomocą komendy:

```
sudo reboot now
```

3.4. Skrypty w języku python

3.4.1. Temperature Producer

Skrypt *Temperature Producer* znajduje się na wykorzystywanym urządzeniu Raspberry Pi. Korzysta on z dwóch bibliotek

- `wlthermsensor` - zawiera metodę do pobrania temperatury z czujnika,
- `kafka` - odpowiada za przysyłanie wiadomości do servera kafki

Na początku należy dokonać definicji producenta wiadomości dla serwera Kafka, przypisujemy adres hosta i port hosta, oraz wybieramy sposób serializowania wiadomości, może być dowolny, lecz należy pamiętać by był on zgodny ze sposobem użytym w konsumerze.

```
1 temperature_producer = KafkaProducer(  
2     bootstrap_server=['host_ip:port'], #change to current host  
3     IP  
4     value_serializer=lambda v: json.dumps(v).encode('utf-8'),  
5     api_version=(3,3,1)  
6 )
```

Listing 1: Definicja producera Kafka

Protokołem wykorzystywanym jest domyślny dla Kafki protokół TCP.

Pętla *While* odpowiadająca za pobranie i wysłanie informacji. Operacja wysłania zostaje zakończona komendą *producer.flush()*. Gwarantującą, że wszystkie wiadomości przygotowane do wysłania trafią do brokera wiadomości kafka.

```
1 kafka_topic = 'Ethernet'  
2  
3 while True:  
4  
5     temperature = sensor.get_temperature()  
6     temperature_producer.send(kafka_topic, temperature)  
7     temperature_producer.flush()  
8     sleep(1)
```

Listing 2: Definicja producera Kafka

3.5. Consumer.py

Kafka Consumer znajduje się na tej samej maszynie która służy za *Kafka Server*. Z tego powodu w skrypcie wykorzystano argument: `bootstrap_servers[localhost:9092]`

Kafka Consumer definiujemy analogicznie do Producera:

```
1 simple_consumer = KafkaConsumer(  
2     auto_offset_reset='earliest',  
3     bootstrap_servers=['localhost:9092'],  
4     api_version=(3,3,1))  
5  
6 simple_consumer.subscribe('Ethernet')
```

Listing 3: Definicja producera Kafka

Operacja kończy się zasubskrybowaniem do tematu z którego docelowo będą zczytywane dane. Parametr *earliest* jest kluczowy, gdyż dzięki niemu możliwe jest odczytywanie wiadomości w miarę jak nadchodzą, co jest funkcjonalnością ostatniej części *Consumera*.

4. Wnioski

Projekt z wykorzystaniem czujnika temperatury, Raspberry Pi i biblioteki Kafka Python dostarczył wartościowych wniosków oraz zapewnił cenne doświadczenie związane z sieciami komputerowymi. Debugowanie błędów połączenia z brokerem Kafka ujawniło problem z DNS Resolution. Rozwiązanie wymagało edycji pliku */etc/hosts*. rojekt dostarczył cenne doświadczenie w zakresie konfiguracji połączenia z brokerem Kafka, produkcji i konsumpcji danych w czasie rzeczywistym oraz rozwiązywania problemów z sieciami komputerowymi.

5. Pliki projektu

<https://github.com/Stefan-Luna-Domaradzki/Kafka-Raspberry-Thermometer>

Literatura

[1] <http://weii.portal.prz.edu.pl/pl/materialy-do-pobrania>. Dostęp 5.01.2015.

[2] <https://forbot.pl/blog/kurs-raspberry-pi-czujnik-temperatury-ds18b20-id26430>

STRESZCZENIE PRACY DYPLOMOWEJ

**AGREGACJA DANYCH Z CZUJNIKA I RASPBERRY PI PRZY
UŻYCIU APACHE KAFKA I JĘZYKA PYTHON**

Autor: Stefan Domaradzki, nr albumu: FS-166642

Opiekun: Dr. Marek Bolanowski

Słowa kluczowe: (max. 5 słów kluczowych w 2 wierszach, oddzielanych przecinkami)

WPISZ-RODZAJ-PRACY THESIS ABSTRACT

TEMAT PRACY PO ANGIELSKU

Author: Stefan Domaradzki, nr albumu: FS-166642

Supervisor:

Key words: (max. 5 słów kluczowych w 2 wierszach, oddzielanych przecinkami)