

# System rozproszonego przetwarzania plików wideo

...

Artur Bauer, Jakub Gajda, Łukasz Kożuszek

# Założenia

- System służy do konwersji plików wideo.
- Użytkownik przesyła do systemu plik, wybiera docelowy format i rozdzielczość, po czym system dokonuje konwersji, dzieląc zadania między zdalne hosty.
- Na koniec użytkownik może pobrać z systemu przekonwertowany plik.

# Przypadki użycia

(1)

Aktor główny: Użytkownik

Aktor pomocniczy: Klasa obsługująca linię poleceń

Opis: Użytkownik przesyła do systemu plik, wybiera docelowy format i rozdzielczość, system powiadamia go o wyniku operacji

# Przypadki użycia

(2)

Aktor główny: Serwer (nadzorczy)

Aktor pomocniczy: Worker (klasa odpowiedzialna za przetworzenie części pliku)

Opis: Serwer dzieli otrzymany od użytkownika plik na części i rozsyła je do rozproszonych Workerów, które następnie konwertują je do zadanego formatu i rozdzielczość

# Przypadki użycia

(3)

Aktor główny: Serwer (nadzorczy)

Aktor pomocniczy: Worker (klasa odpowiedzialna za przetworzenie części pliku)

Opis: Serwer po otrzymaniu komunikatu o gotowości Workera do wysłania pliku po konwersji przyjmuje pliki i scala je w jedną całość

# Przypadki użycia

(4)

Aktor główny: Worker

Aktor pomocniczy: Server

Opis: Worker otrzymuje nowy plik na wewnętrzną kolejkę przetwarzania. Zapamiętane są adresy źródłowe oraz komenda , którą należy wykonać, aby skonwertować plik.

# Komunikacja

- Po uruchomieniu Workerzy zgłaszają się do Serwera (wiadomość JOIN)
- Serwer zapisuje dane każdego nowego Workera i przesyła do niego potwierdzenie zaakceptowania (wiadomość JOIN ACCEPT)
- Co 5 sekund Serwer odpytuje Workerów o liczbę plików, jakie są w danej chwili zdolne przetworzyć (wiadomość FREE SPACE REQUEST)
- Workerzy odpowiadają wiadomością FREE SPACE ANSWER

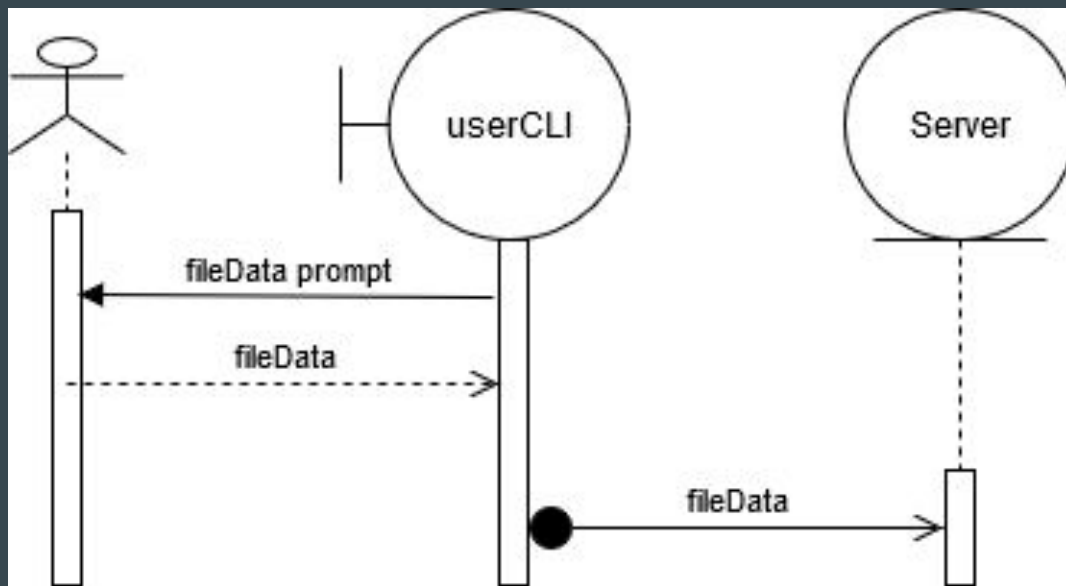
# Komunikacja

- W razie otrzymania nowego pliku od użytkownika Serwer dzieli go na tyle części, ile wynosi suma wartości zadeklarowanych przez Workerów w wiadomościach FREE SPACE ANSWER, jednak co najwyżej na tyle, aby długość każdej części nie wynosiła mniej niż 10 sekund
- Po podzieleniu pliku na części Serwer przesyła je do Workerów (wiadomość CONVERT FILE)
- Serwer co 5 sekund odpytuje Workerów, czy przekonwertowały swoje pliki (wiadomość SEND FILE)
- Jeżeli Worker ukończył konwersję, to odpowiada na tę wiadomość przesłaniem pliku do Serwera (wiadomość CONVERTED)



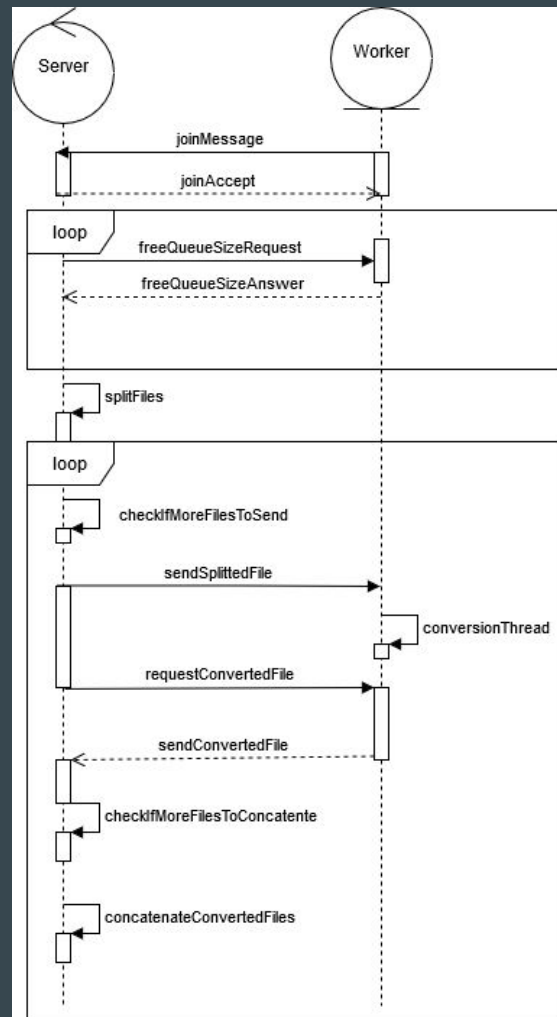
# Diagram sekwencji

Użytkownik wprowadza plik do konwersji



# Diagram sekwencji

Serwer wykonuje konwersję przy użyciu zdalnych węzłów,  
poprzez wymianę wiadomości



# Aplikacja: github

- Aplikacja została napisana w Pythonie, przy użyciu m.in. bibliotek socket, threading, PythonVideoConverter i queue
- Węzły systemu są reprezentowane przez procesy, identyfikowane przy pomocy numerów PID, komunikują się za pośrednictwem socketów
- Dane o plikach przeznaczonych do konwersji oraz tych skonwertowanych są przechowywane w kolejkach FIFO
- Dzielenie, konwersja i łączenie plików jest wykonywana przy pomocy frameworku FFMPEG, do którego API zapewnia biblioteka PythonVideoConverter

# Wymagania aplikacji

Zaleca się użycie python w wersji 3.7 lub wyższej. Wymagane jest zainstalowanie następujących bibliotek:

- PythonVideoConverter

<https://pypi.org/project/PythonVideoConverter/>

Można to wykonać poleceniem `pip install -r requirements.txt`

# Biblioteki python

- Socket – Służy do komunikacji z między Workerami oraz Serverem
- Threading – Utworzenie wątków dla komunikacji z każdym Workerem. W Workerze utworzone są wątki do nasłuchiwania na zapytania Servera oraz dla konwersji plików
- PythonVideoConverter – Nakładka do ffmpeg w celu łatwiejszego używania
- Queue – synchroniczne kolejki plików do przetworzenia oraz przetworzonych zarówno po stronie Workera oraz Servera

# Instrukcja uruchamiania

1. Należy ustawić zmienną środowiskową z ścieżką do ffmpeg
  - a. `export FFMPEG_PATH = "/usr/bin/ffmpeg;/usr/bin/ffprobe"`
2. Należy uruchomić Server
  - a. `python Server.py`
3. Należy uruchomić Workery (w zależności od potrzeb, minimum 1)
  - a. `python Worker.py`
4. Należy podać nazwę pliku, rozszerzenie, rozdzielczość oraz lokalizację zapisu w Serwerze

```
Please enter file path BigBuckBunny.mp4
Please enter your desired extension. You may choose between: mp4, avi, webm, wmv avi
Enter your desired width 100
Enter your desired height 100
Enter location where your file will be saved after conversion .
```

# Dziękujemy za uwagę

Artur Bauer, Jakub Gajda, Łukasz Kożuszek