

# Systemy rozproszone - laboratorium 2

Łukasz Jezapkowicz - 305313

10 marca 2021

# 1 Laboratorium 2

Ten dokument zawiera odpowiedzi do zadań z laboratorium 2 z Systemów Rozproszonych.

## 1.1 Zadanie 1a - Zaobserwuj działanie mechanizmu niezawodności w różnych scenariuszach potwierdzeń wiadomości.

### 1.1.1 Który sposób potwierdzeń zapewnia większą niezawodność?

Dla scenariusza 1 (zrestartowanie konsumenta po zakończeniu przetwarzania wiadomości) każdy sposób jest tak samo niezawodny, ponieważ wiadomość została w pełni przetworzona i żadne dane nie zostaną utracone. Dla scenariusza 2 (zrestartowanie konsumenta podczas przetwarzania wiadomości) oczywistą przewagę ma potwierdzanie po przetworzeniu wiadomości. Jeżeli przetwarzanie wiadomości zajmuje pewien okres czasu (tutaj `Thread.sleep`) to dla potwierdzania przy odbiorze nasza wiadomość nie wykona się do końca. Dla przypadku potwierdzania po przetworzeniu wiadomości wiadomość będzie przetwarzana od początku aż nie zostanie potwierdzona (można resetować wiele razy). Podczas testów zauważyłem również, że dla potwierdzania przy odbiorze, przy przerwaniu, gubią się kolejne wiadomości (nie jestem pewien czy taki efekt jest poprawny).

### 1.1.2 Co się stanie, jeśli nie będziemy potwierdzać wiadomości ani po otrzymaniu, ani po przetworzeniu?

Jeżeli nie będziemy potwierdzać otrzymania wiadomości to Broker nie ma pojęcia czy wiadomość faktycznie dotarła do konsumenta. Ciągnie to za sobą wiele konsekwencji, ponieważ nie jesteśmy w stanie wykryć błędu (zła kolejka, zły topic itp.) i nasza komunikacja staje się o wiele bardziej podatna na błędy a co za tym idzie o wiele mniej użyteczna.

## 1.2 Zadanie 1b - Zaobserwuj działanie mechanizmu load-balancing.

### 1.2.1 Przed włączeniem obsługi QoS

```
Z1 CONSUMER
Waiting for messages...
Received: 1
Sleeping for 1 seconds.
Sleeping done.
Received: 1
Sleeping for 1 seconds.
Sleeping done.
Received: 1
Sleeping for 1 seconds.
Sleeping done.
Received: 1
Sleeping for 1 seconds.
Sleeping done.
Received: 1
Sleeping for 1 seconds.
Sleeping done.
```

```
Z1 CONSUMER
Waiting for messages...
Received: 5
Sleeping for 5 seconds.
Sleeping done.
Received: 5
Sleeping for 5 seconds.
Sleeping done.
Received: 5
Sleeping for 5 seconds.
Sleeping done.
Received: 5
Sleeping for 5 seconds.
Sleeping done.
Received: 5
Sleeping for 5 seconds.
Sleeping done.
```

### 1.2.2 Po włączeniu obsługi QoS

```
Z1 CONSUMER
Waiting for messages...
Received: 1
Sleeping for 1 seconds.
Sleeping done.
Received: 1
Sleeping for 1 seconds.
Sleeping done.
Received: 5
Sleeping for 5 seconds.
Sleeping done.
Received: 1
Sleeping for 1 seconds.
Sleeping done.
Received: 5
Sleeping for 5 seconds.
```

```

Z1 CONSUMER
Waiting for messages...
Received: 5
Sleeping for 5 seconds.
Sleeping done.
Received: 1
Sleeping for 1 seconds.
Sleeping done.
Received: 5
Sleeping for 5 seconds.
Sleeping done.
Received: 1
Sleeping for 1 seconds.
Sleeping done.
Received: 5
Sleeping for 5 seconds.
Sleeping done.

```

Jak widać po włączeniu QoS (load balancing) nasi konsumenci nie działają zgodnie z algorytmem round-robin (na zmianę, co powodowało, że jeden konsument dostawał "cięższe" wiadomości). Teraz wiadomości przetwarza ten konsument, który w danej chwili jest wolny, dzięki czemu zadania rozkładają się lepiej pośród konsumentów.

## 1.3 Zadanie 2 - Zaimplementuj oraz pokaż w działaniu routing Direct oraz Topic.

### 1.3.1 Zasada działania routingu Direct

Routing Direct pozwala na związanie kolejki z kluczem/kluczami. Producent podczas wysyłania wiadomości określa klucz wiadomości (np. orange). Wiadomość trafia następnie do Exchange (o typie Direct), który rozsyła wiadomość na wszystkie kolejki, które związały się z takim kluczem. Wiele kolejek może mieć ten sam klucz oraz kolejka może mieć wiele kluczy.

Przykład:

Kolejka 1 o kluczach: orange, black, green

Kolejka 2 o kluczach: green, white, pink

Wysłane wiadomości (wiadomość — klucz):

green — green

black — black

orange — orange

white — white

pink — pink

yellow — yellow

Kolejka 1 powinna otrzymać wiadomości: green, black, orange.

Kolejka 2 powinna otrzymać wiadomości: green, white, pink.

```
Z2 CONSUMER
Please provide routing keys (separated by space): orange black green
created queue: amq.gen-iA4uUcvMx-TVnWqAxl5_A
Waiting for messages...
Received: green
Received: black
Received: orange
```

Kolejka 1

```
Z2 CONSUMER
Please provide routing keys (separated by space): green white pink
created queue: amq.gen-1U1DGroimcksMwg4MvFBng
Waiting for messages...
Received: green
Received: white
Received: pink
```

Kolejka 2

Jak widać każda kolejka otrzymała tylko wiadomości ze zgodnym kluczem.

### 1.3.2 Zasada działania routingu Topic

Routing Topic pozwala na związanie kolejki z wzorcem (np. `.monkey.*`). Producent podczas wysyłania wiadomości określa klucz wiadomości (np. `monkey.banana`). Wiadomość trafia następnie do Exchange (o typie Topic), który rozsyła wiadomość na wszystkie kolejki, których wzorce pasują do podanego klucza. Wiele kolejek może mieć ten sam wzorec oraz kolejka może mieć wiele wzorców.

Przykład:

Kolejka 1 o wzorcach: `#.monkey.*`, `*.burger.*`

Kolejka 2 o wzorcach: `*.monkey.#`, `burger.#`

Wysłane wiadomości (wiadomość — klucz):

`monkey.banana` — `monkey.banana`

`banana.monkey` — `banana.monkey`

`monkey` — `monkey`

`banana.monkey.banana` — `banana.monkey.banana`

`burger` — `burger`

`yummy.burger` — `yummy.burger`

`burger.yummy` — `burger.yummy`

`yummy.burger.yummy` — `yummy.burger.yummy`

Kolejka 1 powinna otrzymać wiadomości: `monkey.banana`, `banana.monkey.banana`, `yummy.burger.yummy`.

Kolejka 2 powinna otrzymać wiadomości: `banana.monkey`, `banana.monkey.banana`, `burger`, `burger.yummy`.

```
Z2 CONSUMER
Please provide routing keys (separated by space): #.monkey.* *.burger.*
created queue: amq.gen-5Ihvwn4fJb-REjFBY9vvJw
Waiting for messages...
Received: monkey.banana
Received: banana.monkey.banana
Received: yummy.burger.yummy
```

## Kolejka 1

```
Z2 CONSUMER
Please provide routing keys (separated by space): *.monkey.# burger.#
created queue: amq.gen-uNU7wUETn1nMDxJHcrI7Hg
Waiting for messages...
Received: banana.monkey
Received: banana.monkey.banana
Received: burger
Received: burger.yummy
```

## Kolejka 2

Jak widać każda kolejka otrzymała tylko wiadomości ze zgodnym kluczem.

### 1.3.3 Różnica w działaniu

Różnica między działaniem routingu Direct oraz Topic jest łatwo zauważalna. Przy routingu Direct wiadomość dotrze do kolejki tylko jeżeli klucze są identyczne zaś dla routingu Topic wystarczy, żeby klucz pasował do pewnego wzorca. To sprawia, że routing Topic ma szersze zastosowanie (routing Direct jest odmianą routingu Topic). Możemy dla niego na przykład wyszukiwać słowa kluczowe takie jak soccer, computer, java.