

Raport RabbitMQ, Systemy Rozproszone 2020

Karol Lasocki 296610

1a.

Potwierdzenie	Restart po przetworzeniu wiadomości	Restart w trakcie przetwarzania wiadomości
Po otrzymaniu wiadomości	Tak	Nie
Po przetworzeniu wiadomości	Tak	Tak

- Który sposób potwierdzeń zapewnia większą niezawodność?

Potwierdzanie po przetworzeniu wiadomości - dzięki temu mamy pewność że wiadomości nie zostaną "zgubione" przy restarcie, awarii itp. Ten sposób ma jednak poważną wadę - wymaga obszernej obsługi błędów, bo jeśli wiadomość powoduje błąd w systemie, w naszym przykładzie - nie jest liczbą, konsument będzie próbował ją przetworzyć do skutku i w efekcie nie będzie w stanie przetworzyć żadnych innych, nawet poprawnych wiadomości.

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

RabbitMQ będzie je ciągle przechowywał, próbując dostarczać ponownie i czekając na potwierdzenie, pożerając coraz więcej pamięci

1b.

Przed włączeniem QoS:

Consumer1:

Waiting for messages...

Received: 1

Received: 1
Received: 1
Received: 1
Received: 1
Consumer2:
Waiting for messages...
Received: 5
Received: 5
Received: 5
Received: 5
Received: 5

Po włączeniu Qos:
Consumer1:
Waiting for messages...
Received: 1
Received: 1
Received: 5
Received: 1
Received: 5

Consumer2:
Waiting for messages...
Received: 5
Received: 1
Received: 5
Received: 1
Received: 5

2.

Routing direct służy do rozdzielania wiadomości dla kolejek na podstawie dokładnej wartości klucza - oznacza to że zbiory odbieranych wiadomości muszą być rozłączne, lub się pokrywać. Przykładowa komunikacja:

Z2 PRODUCER
Enter binding key:
animals
Enter message:
This is for all animal owners!

Sent: This is for all animal owners!

Enter binding key:

dogs

Enter message:

This is for all dog owners!

Sent: This is for all dog owners!

Enter binding key:

Z2 CONSUMER

Enter binding key:

animals

created queue: amq.gen-SEaoqiroPUEJugXWV0SSyA

Waiting for messages...

Received: This is for all animal owners!

Z2 CONSUMER

Enter binding key:

dogs

created queue: amq.gen-luZgjkxfgtTEZ5BaqSwZrw

Waiting for messages...

Received: This is for all dog owners!

Widać problem związany z tym rodzajem routingu - nie da się zamodelować przypadku gdy zbiory mają część wspólną, lub są podzbiorami (dogs podzbiorem animals). W takich sytuacjach świetnie sprawdza się routing Topic, z dwoma specjalnymi symbolami: * - dowolne słowo, # - 0 lub więcej dowolnych słów. Przykład:

Z2 PRODUCER

Enter binding key:

animals.dogs.blue.pretty

Enter message:

This is for pretty blue dogs!

Sent: This is for pretty blue dogs!

Enter binding key:

animals.cats

Enter message:

This is for all cats!

Sent: This is for all cats!

Enter binding key:

animals.cats.blue.furry

Enter message:

This is for furry blue cats!

Sent: This is for furry blue cats!

Enter binding key:

Z2 CONSUMER

Enter binding key:

*.cats.#

created queue: amq.gen-4g9UeluYH2UpGMbFQ0ZeAg

Waiting for messages...

Received: This is for all cats!

Received: This is for furry blue cats!

Z2 CONSUMER

Enter binding key:

animals.*.blue.#

created queue: amq.gen-fBHWNQ2D-09CDUc2rdWsLQ

Waiting for messages...

Received: This is for pretty blue dogs!

Received: This is for furry blue cats!