



Образование для инженеров
и технических лидеров

RabbitMQ - Дополнительные ВОЗМОЖНОСТИ

Спикер: Алексей Барабанов



Цель урока

Узнать о дополнительных возможностях

План урока

- **Расчет количества консьюмеров** - по формуле Эрланга
- **Шардирование** - различные способы балансировки
- **Дедупликация** сообщений в очереди
- **Многоуровневая очередь** повторных попыток
- **Приоритеты** сообщений
- **Практическое задание**

Агнер Краруп Эрланг 1878-1929



Формула Эрланга А. К. 1917г



Erlang C formula

$$P_w = \frac{\frac{A^N}{N!} \frac{N}{N - A}}{\left(\sum_{i=0}^{N-1} \frac{A^i}{i!} \right) + \frac{A^N}{N!} \frac{N}{N - A}}$$

Формула Эрланга А. К. 1917г

Google Калькулятор Эрланга с

Все Картинки Новости Покупки Видео Ещё Инструменты

Результатов: примерно 6 120 (0,43 сек.)

<https://www.mightycall.ru> > kalkulyator-erlanga-s

Калькулятор Эрланга: расчет производительности колл ...

Калькулятор Эрланга С: Расчёт количества количества операторов колл-центра с учетом количества звонков, средней продолжительностью разговора и среднего ...

<https://callcenter.by> > erlang

Калькуляторы Эрланга, расчет нагрузки на колл-центры

Калькулятор Erlang С. Данный калькулятор, основанный на модели расчета нагрузки Erlang С, позволяет вычислить количество операторов, которые должны работать в ...

<https://astrapage.ru> > calculator

Калькулятор Эрланга С - Астра Пейдж

Калькулятор Эрланга С. Калькулятор (формула) Эрланга С – эффективный инструмент для расчета количества операторов в линии в зависимости от нагрузки для ...

<http://globalhome.su> > content > article > erlang_c

Расчет количества операторов. Калькулятор Erlang С

Калькулятор Erlang С. С помощью данного калькулятора можно вычислить необходимое количество операторов и другие параметры call-центра в зависимости от ...

Формула Эрланга А. К. 1917г

<https://gitlab.slurm.io/rabbitmq/erlang-c>

calls Кол-во за интервал
interval, sec Интервал, с (900=15 мин)
maxq Максимальная длина очереди
call time, sec Время обработки
answer time, sec Время ожидания
percent SLA 1=100%

Calculate!

Consumers count: 18

Онлайн сервисы

ERLANG C

Среднее время разговора, сек:

5

Среднее время пост-обработки звонка, сек:

5

Число звонков в час:

4000

Средняя задержка при ответе на звонок, сек:

1

Рассчитать число операторов

| Количество операторов | Вероятность соединения с оператором без очереди | Среднее время ожидания, сек. | Средняя длина очереди | Уровень обслуживания * | Занятость операторов |
|-----------------------|---|------------------------------|-----------------------|------------------------|----------------------|
| 12 | 14% | 10 | 11 | 21% | 93% |
| 13 | 35% | 3 | 4 | 46% | 85% |
| 14 | 55% | 2 | 2 | 66% | 79% |
| 15 | 71% | 1 | 1 | 80% | 74% |
| 16 | 82% | 0 | 0 | 89% | 69% |
| 17 | 89% | 0 | 0 | 94% | 65% |
| 18 | 94% | 0 | 0 | 97% | 62% |

* Количество звонков, ожидавших меньше заданного интервала (1 сек)

Шардирование

- Цель - балансировка сообщений по очередям/инстансам
- **Плагин x-consistent-hash** - консистентное шардирование
- **Плагин x-modulus-hash** - шардирование по целочисленному делению
- **Плагин x-random** - случайное шардирование
- Можно настроить **шардирование shovel**'ами

Включаем плагин x-consistent-hash

- Содержимое файла **enabled_plugins**

```
[rabbitmq_consistent_hash_exchange,rabbitmq_management].
```

Создаём exchange

- Новый вид exchange

▼ Add a new exchange

Name: *

Type:

Durability:

Auto delete: ?

Internal: ?

Arguments: =

Add **Alternate exchange** ?

Подключаем очереди

- **Routing-key** = вес при распределении

▼ Bindings

This exchange

⇓

| To | Routing key | Arguments | |
|----|-------------|-----------|--------|
| q1 | 1 | | Unbind |
| q2 | 1 | | Unbind |
| q3 | 1 | | Unbind |
| q4 | 3 | | Unbind |

Шардирование по header x-consistent-hash

- Можно шардировать по header'у

▼ **Add a new exchange**

Name: *

Type: ⌵

Durability: ⌵

Auto delete: ? ⌵

Internal: ? ⌵

Arguments: =
 =

Add **Alternate exchange** ?

Add exchange

Включаем плагин x-modulus-hash

- Содержимое файла **enabled_plugins**

```
[rabbitmq_sharding,rabbitmq_management].
```

Создаём exchange x-modulus-hash

- Новый вид exchange x-modulus-hash

▼ Add a new exchange

Name: *

Type: ▾

Durability: ▾

Auto delete: ? ▾

Internal: ? ▾

Arguments: =

Add **Alternate exchange** ?

Add exchange

Подключаем очереди

- **Routing-key** - игнорируется

▼ Bindings

This exchange

⇓

| To | Routing key | Arguments | |
|----|-------------|-----------|--------|
| q1 | | | Unbind |
| q2 | | | Unbind |
| q3 | | | Unbind |
| q4 | | | Unbind |

Включаем плагин x-random

- Активировать плагин можно как через `enabled_plugins` - так и

без перезагрузки командой:

```
rabbitmq-plugins enable rabbitmq_random_exchange
```

Включаем плагин x-random

- Новый вид exchange - **x-random**

▼ Add a new exchange

Name: *

Type: ▾

Durability: ▾

Auto delete: ? ▾

Internal: ? ▾

Arguments: =

Add **Alternate exchange** ?

Add exchange

Биндим очереди

- **Routing-key** - игнорируется

▼ Bindings

This exchange

⇓

| To | Routing key | Arguments | |
|----|-------------|-----------|--------|
| r1 | | | Unbind |
| r2 | | | Unbind |
| r3 | | | Unbind |
| r4 | | | Unbind |

Балансировка через shovel

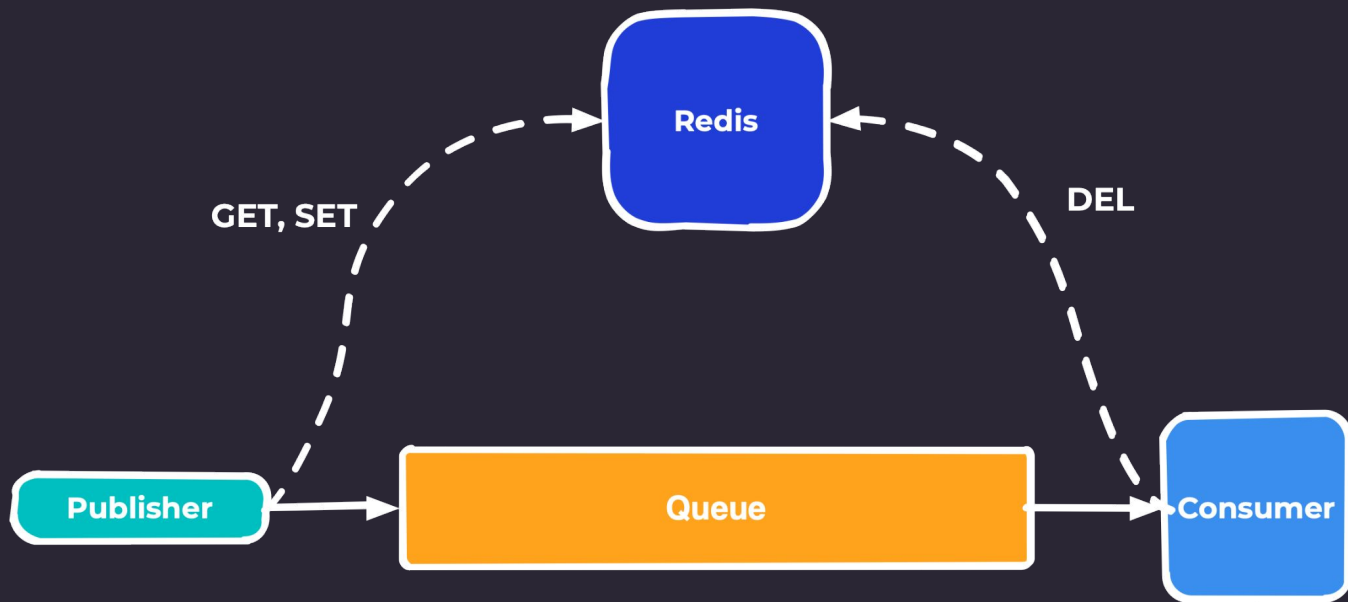
- **Без ключа шардирования**, просто случайным образом

Shovel Status

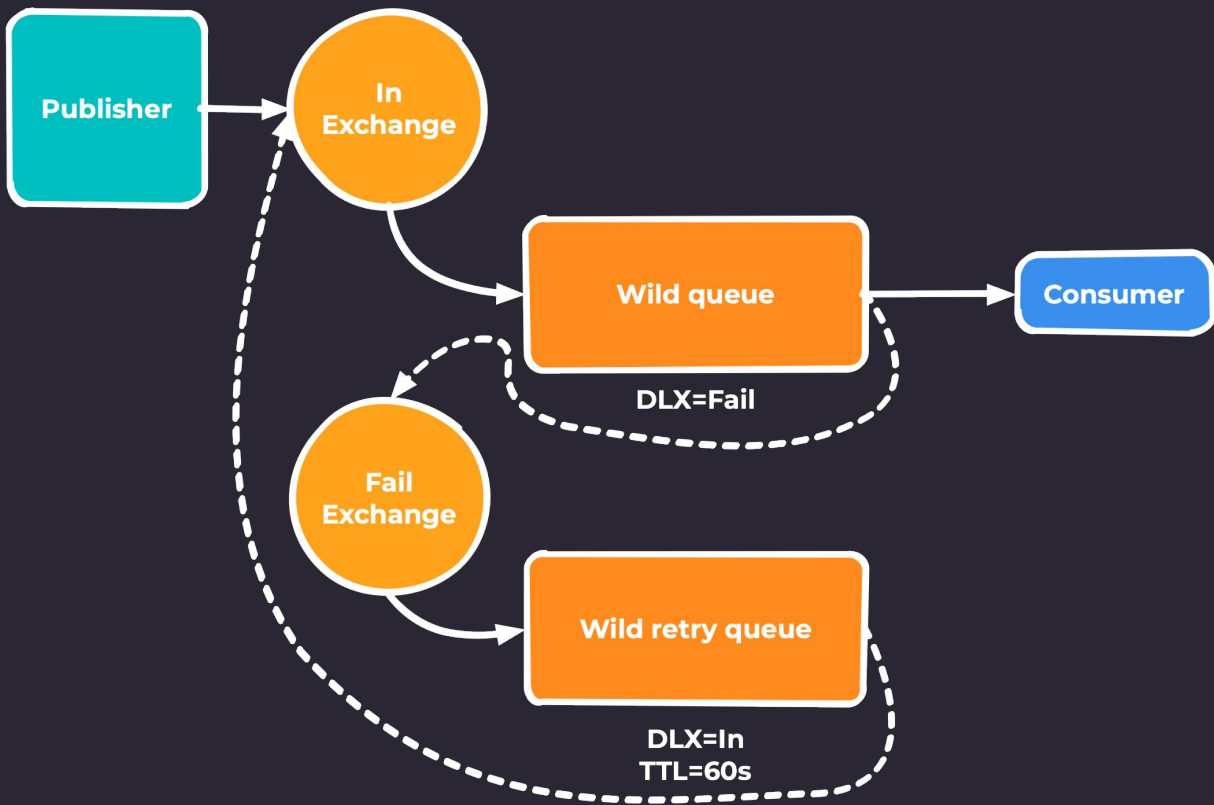
| Name | State | Source | | | Destination | | | Last changed | Operations |
|---------------|-----------|---------|---------|-------------|-------------|---------|-------------|--------------------|------------|
| s1 dynamic | ■ running | amqp091 | amqp:// | in queue | amqp091 | amqp:// | s1 queue | 2022-11-07 9:35:04 | Restart |
| s2 dynamic | ■ running | amqp091 | amqp:// | in queue | amqp091 | amqp:// | s2 queue | 2022-11-07 9:35:48 | Restart |
| s3 dynamic | ■ running | amqp091 | amqp:// | in queue | amqp091 | amqp:// | s3 queue | 2022-11-07 9:35:56 | Restart |
| s4 dynamic | ■ running | amqp091 | amqp:// | in queue | amqp091 | amqp:// | s4 queue | 2022-11-07 9:36:09 | Restart |

Дедупликация сообщений в очереди

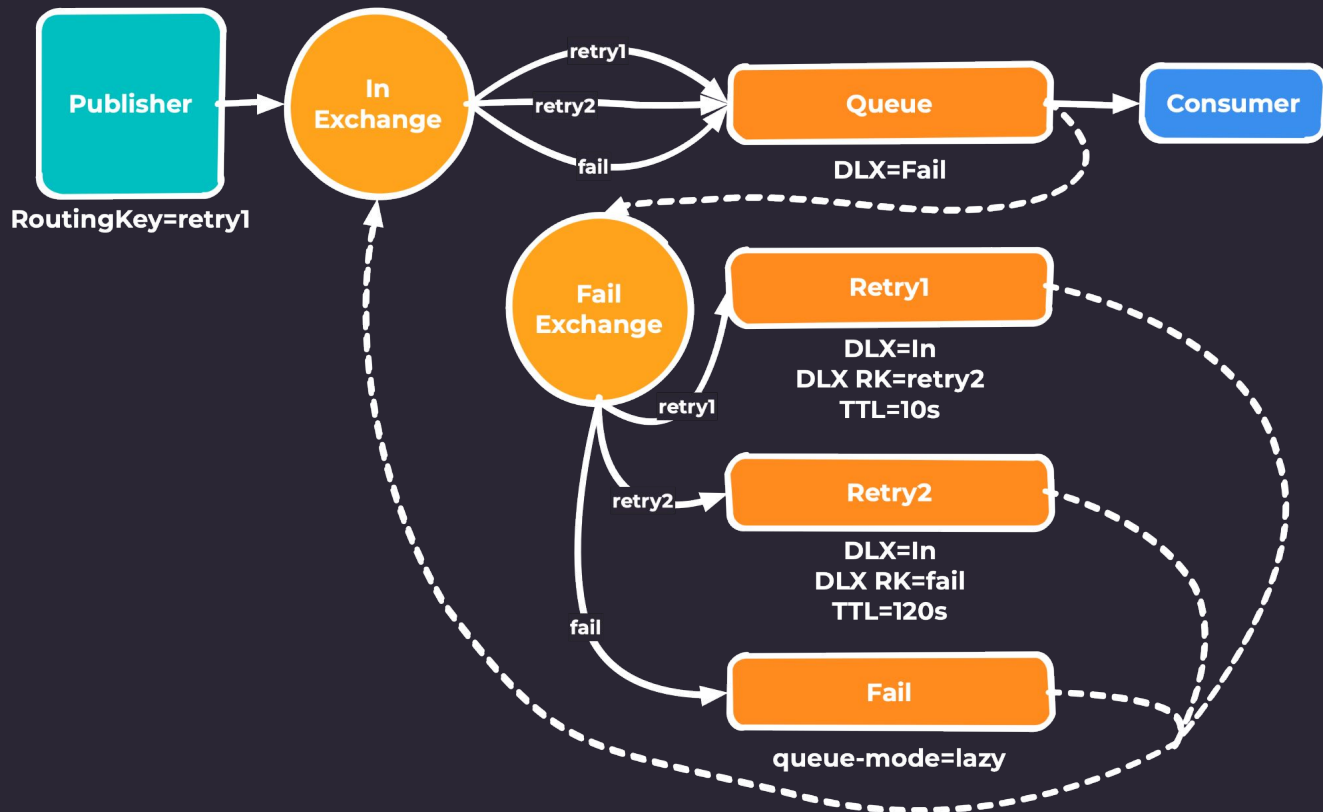
- Штатного механизма дедупликации у RabbitMQ нет



Многоуровневая очередь ПП



Многоуровневая очередь ПП



Приоритеты сообщений

▼ Add a new queue

Type: Classic

Name: priority *

Durability: Durable

Auto delete: ? No

Arguments: x-max-priority = 5 Number

= String

Add Auto expire ? | Message TTL ? | Overflow behaviour ?

Single active consumer ? | Dead letter exchange ? | Dead letter routing key ?

Max length ? | Max length bytes ?

Maximum priority ? | Lazy mode ? | Version ? | Master locator ?

Add queue

Приоритеты сообщений

▼ Publish message

Message will be published to the default exchange with routing key **priority**, routing it to this queue.

Delivery mode: 1 - Non-persistent

Headers: ?

= String

Properties: ?

priority = 5

=

Payload:

5

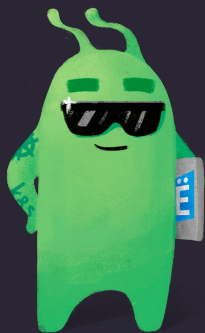
Payload encoding:

String (default)

Publish message

Конец

А кто слушал - молодец



Практическое задание

- Запустить **RabbitMQ** с сохранением стейта
- Включить все плагины шардирования **persistent-hash, mod, random**
- Создать exchange всех типов шардирования, создать и подключить очереди
- **Проверить работу шардирования** - отправкой сообщений
- Настроить и проверить **многоволновую очередь повторных попыток**