Reducing load with RabbitMQ

Kaunas PHP v.28

I’m Povilas Balzaravičius

- PHP developer over 10 years
- Software Engineer at Uber
- Before: developer at Boozt.com (Estina)
- Organizer of Vilnius PHP UG
- Not professional speaker but professional developer :-)

What will be this talk about?

- The Problem
- RabbitMQ fundamentals
- How Problem was solved
- How we use messaging in general

EASY RETURNS IN 30 DAYS

CUSTOMER SERVICE

Boozt.com
WE DELIVER FASHION

WOMEN MEN KIDS SHOES & BAGS DESIGNERS BRANDS A-Z

51 3

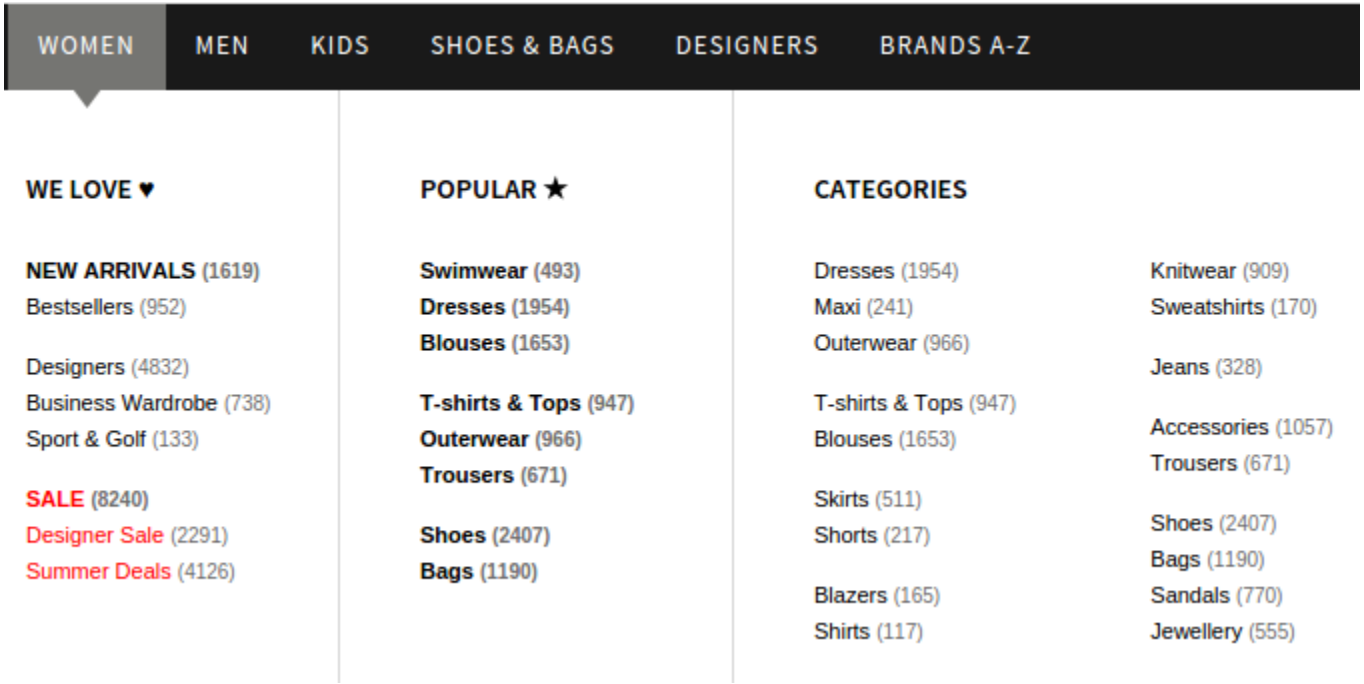
What is Boozt.com?

- One of top clothes e-store in Scandinavia
- Over 300 brands & 10000 styles online
- Over 700 categories

The Problem

(or task)

Many items in categories



Many items in categories

Possibly products at the top of page are selling better.

When customer navigates to category page,
how products must be ordered?

Many items in categories

Items can not be sorted:

- Manually
- Randomly (or how they are stored on db)
- By one of sorting options (price, arrival date, items on sale, etc.)

Autosort

was implemented

What is Autosort?

- Splits page into **segments** by number of items
- Each segment defined by **rules**
- Segments can be sorted and shuffled
- Collection of segments is called **“Autosort”**
- “Autosorts” are assigned to categories

777 styles

Filter by

Sort by

1 / 6

Converse
95.00 €

JUST IN

Clarks
120.00 €

JUST IN

Clarks
130.00 €

JUST IN

GANT
99.00 €

JUST IN

Converse
95.00 €

JUST IN

Clarks
130.00 €

JUST IN

ECCO
120.00 €

+ colors

JUST IN

Havaianas
49.00 €

+ colors

JUST IN

Havaianas
24.00 €

+ colors

Havaianas
34.00 €

Havaianas
34.00 €

+ colors

Havaianas
39.00 €

Content managers ♥ Autosort...

- Segment sizes shrinked from 200 to 4-20 products - more queries
- Implemented autosort inheritance for categories - increased sorted categories count 20-40 times
- Job time increased from 0.25-0.5 to 6-8 hours. Here was our problem :-)

Looking for bottlenecks

- Optimized few code parts - performance increased ~10%
- Tried tune MySQL indexes - no significant impact

Bottleneck found!

Saving products positions to database:

```
UPDATE `product_category` SET `position` = CASE
    WHEN product_id = 123 THEN 1
    WHEN product_id = 456 THEN 2
    WHEN product_id = 789 THEN 3
    ...
    END
WHERE category_id = 1001 AND product_id IN (123, 456, 789)
```

Getting rid of bottleneck

- Update performed on single category only
- Other categories can be sorted during update
- Need parallelize sorting[1]!

[1] With RabbitMQ of course!


RabbitMQ™

RabbitMQ?

- Multi Protocol Messaging Server
- Open source
- Commercial support
- Messaging via AMQP (Advanced Message Queuing Protocol)
- Supports many programming languages

Why we need messaging?

- Separates Job request from the Job
- Jobs can be performed on separate processes and machines
- Workers can be added or removed easily
- Messaging is asynchronous

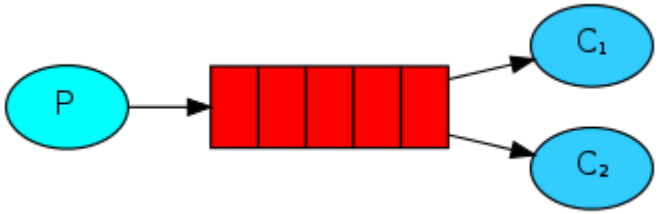
RabbitMQ Basics

- **Producer** - sends messages
- **Consumer** - Running process and waiting for messages
- **Queue** - a buffer that stores messages
- **Exchange** - distributes messages

Simple queue example

1. Producer sends message to named queue
2. Message is stored in queue
3. Consumer takes message from named queue
4. Message removed from queue immediately
5. Consumer “consumes” the message

Consumer becoming slow?



- Just run additional consumer
- Round-robin dispatching by default
- Jobs are processed in parallel!

What happens if consumer dies?

- Message is lost if worker dies
- The task should be delivered to another worker
- RabbitMQ supports message ACKs
- Message is removed when consumer sends ACK
- ACKs are disabled by default

What happens if server dies?

- Messages in queues will be lost
- RabbitMQ supports **durable** queues
- Queue needs to be re-created as durable if was defined before
- Messages should be sent in “persistent” mode
- RabbitMQ doesn't do fsync(2) for every message - not 100% durable :-)

Fair dispatch

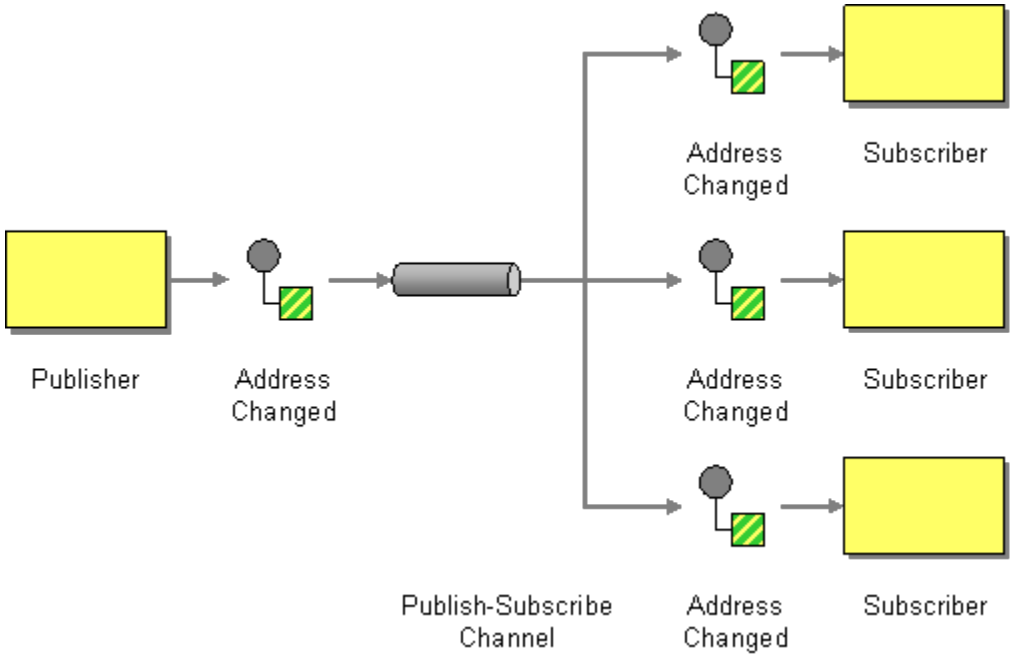
- If consumer C1 will get more complex jobs than C2, C1 will be busier all the time.
- Set ***prefetch_count = 1*** property to not to give more than one message to a worker at a time.
- Requires *ACKing*

Let’s think about logging system

This will help us understand examples

Publish & Subscribe pattern

Deliver same message for multiple consumers



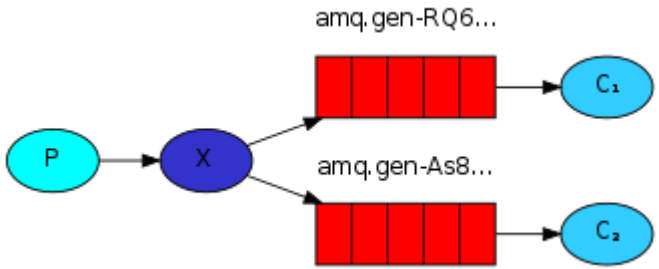
RabbitMQ Exchanges

- RabbitMQ dogma - producer never sends any messages directly to a queue
- Producer sends messages to an exchange
- A **routing_key** must be used with messages
- Exchange type describes behavior:
  - Append the message to one/few/all queues
  - Discard the message
  - Types: fanout, direct, topic, headers

Temporary Queues

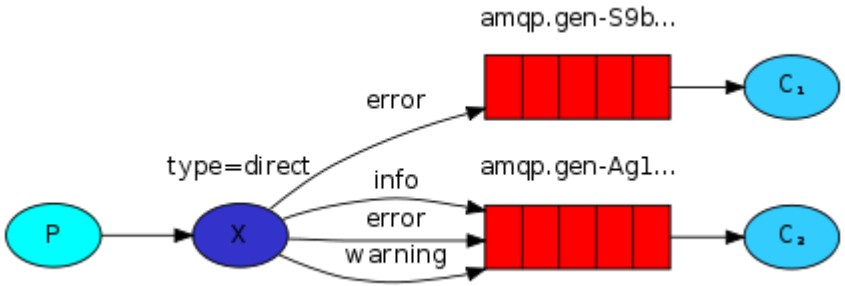
- If no queue is bounded to exchange, messages will be lost.
- Temporary queues are deleted after consumer disconnects.
- Created by consumer when only new messages should be retrieved.
- RabbitMQ supports temporary queues.

Listening all messages: Fanout



- **Fanout** exchange type - send messages to every connected queue
- **Temporary queues** used
- **Routing key** ignored by fanout

Listening messages subset: Direct



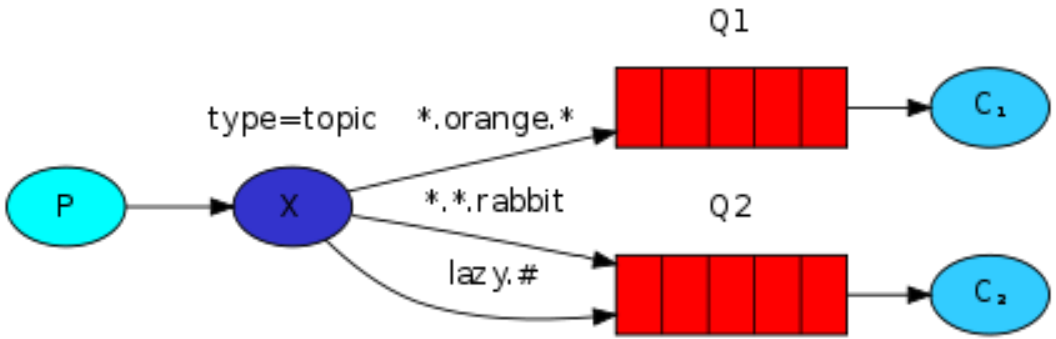
- **Direct** exchange type - deliver msg to queues with same **routing key**
- Routing key not ignored anymore
- Multiple **bindings** can be defined for same queue
- Multiple queues can use same binding key

Listening messages by Topic

- Topic exchange type - filter messages by patterns
- Topic is a list of words separated.by.dots
- Subscribing topic patterns:
  - * (asterisk) substitute one word - cron.*.error
  - # (hash) substitute 0 or more words - cron.#
- Topic exchange can behave as fanout or direct

Listening messages by Topic

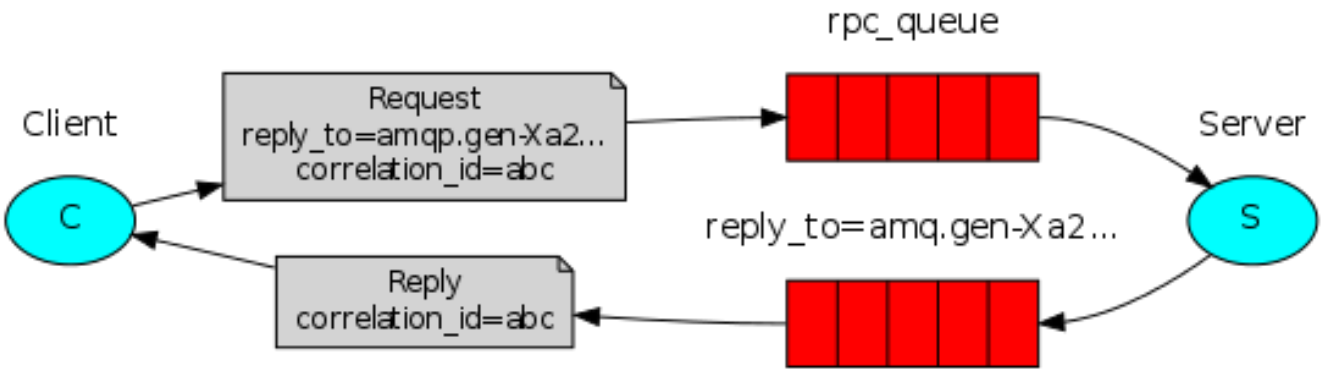
Topic pattern: <speed>.<colour>.<species>



Receiving results: RPC

- Create and provide **callback queue** from provider
- Response will be sent to callback queue
- If single callback queue is used per provider, use **correlation_id** to match request with response
- It is slow and blocking

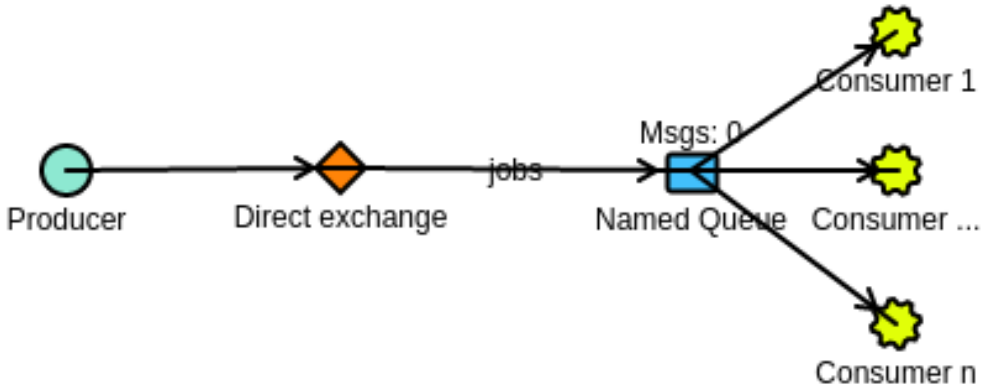
Receiving results: RPC



Now you are Ready to work with RabbitMQ

Let’s get back to Autosort

Scaling Autosort



Scaling Autosort

- Producer sends categories which needs to be sorted
- Exchange type is **direct**
- Single **named queue** with **ACK** and **fair dispatch** is used
- Consumers are calling CLI commands (can be used for other tasks too)

Scaling achievements

BEFORE

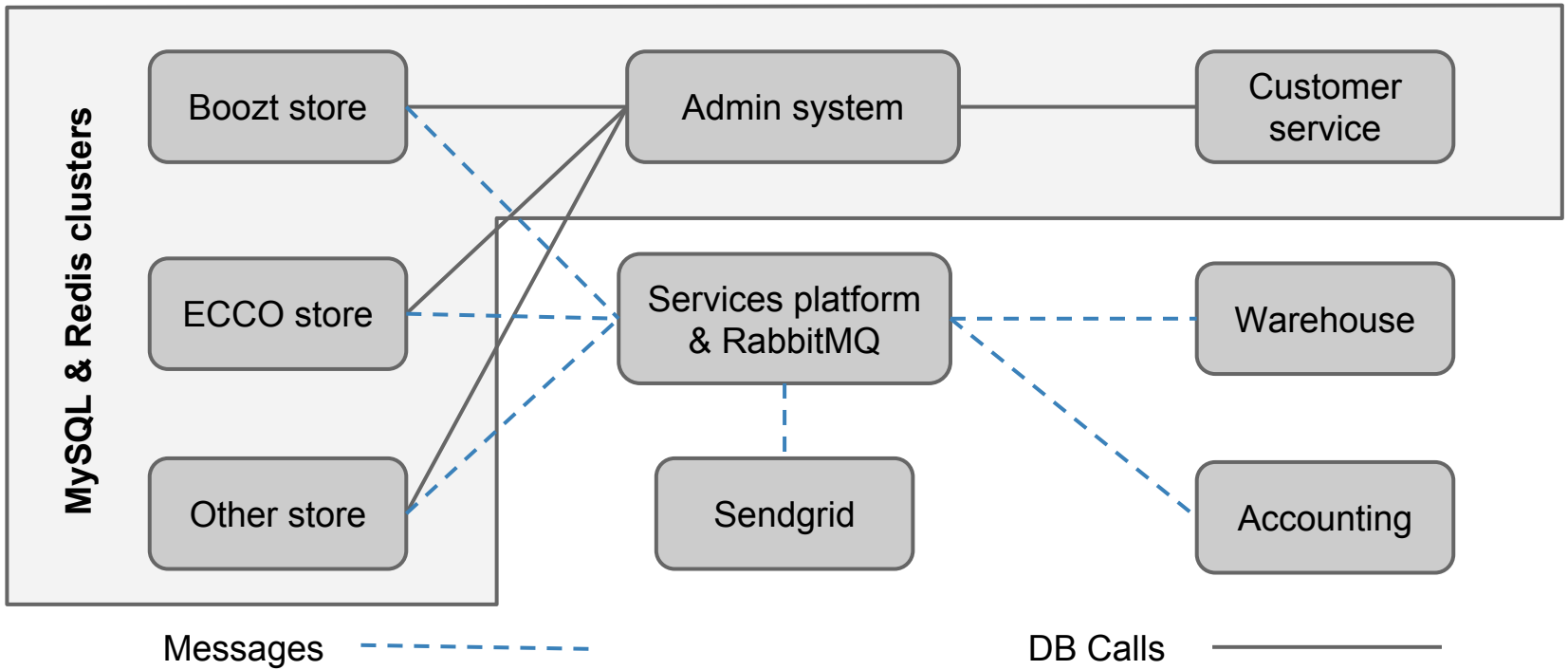
- 6-8 hours
- Single machine
- Single table for all stores

AFTER

- 2-3 hours
- Separate machines
- 10 separate stores!

~28 times faster!

RabbitMQ in Boozt.com



Messaging use case #1

Order completed on **Store**

1. Message sent to **Services** platform from **Store**
2. **Services** sends message to:
    a. **Warehouse** to mark products as reserved
    b. **Accounting** to register an order
    c. **Sendgrid** to send an email

Messaging use case #2

Order confirmed on **Accounting** system

1. Message sent to **Services** from **Accounting**
2. **Services** sends message to:
    a. **Store** to mark order as confirmed
    b. **Sendgrid** to send an email

RabbitMQ ♥ PHP = php-amqplib

- Created by RabbitMQ developer - **Alvaro Videla**
- Install via composer **videlalvaro/php-amqplib** or

https://github.com/videlalvaro/php-amqplib

Dark side of php-amqplib

```php
<?php
// ...
$channel->exchange_declare(
    'topic_logs', 'topic', false, false, false);

list($queue_name, ,) = $channel->queue_declare(
    "", false, false, true, false);
// Any comments required?
```

How to make sure my consumers are running?

Supervisor

- Supervisor is a client/server system that allows its users to monitor and control a number of processes on UNIX-like operating systems.
- Observes if consumers are running and reloads them if they died

/etc/supervisor/conf.d/demo.conf

```
[program:demo]
command=/usr/bin/php /home/pawka/Desktop/rabbitmq-tutorials/php/receive.php
process_name=%(program_name)s
numprocs=1
directory=/tmp
umask=022
priority=999
autostart=true
autorestart=true
stdout_logfile=/tmp/worker.log
user=pawka
```

Tips for messaging

- Think about using unified workers
- And use priority queues for important tasks (available since v3.5.0)
- Workers as P2P clients
- Use correlation_id for messages

Tips for messaging

- Manage your workers with supervisor or similar tool
- Set lifetime for workers (eg. 15 mins)
- Use separate exchanges for separate apps

Resources

- https://www.rabbitmq.com - official site and great tutorial
- http://tryrabbitmq.com - RabbitMQ simulator
- https://github.com/rabbitmq/rabbitmq-tutorials/ - Source code of examples
- http://supervisord.org - supervisor

?

Thank You!

twitter.com/@Pawka github.com/Pawka