Otto Pagels Fick

Report: *Lab 3 RabbitMQ*

software architecture

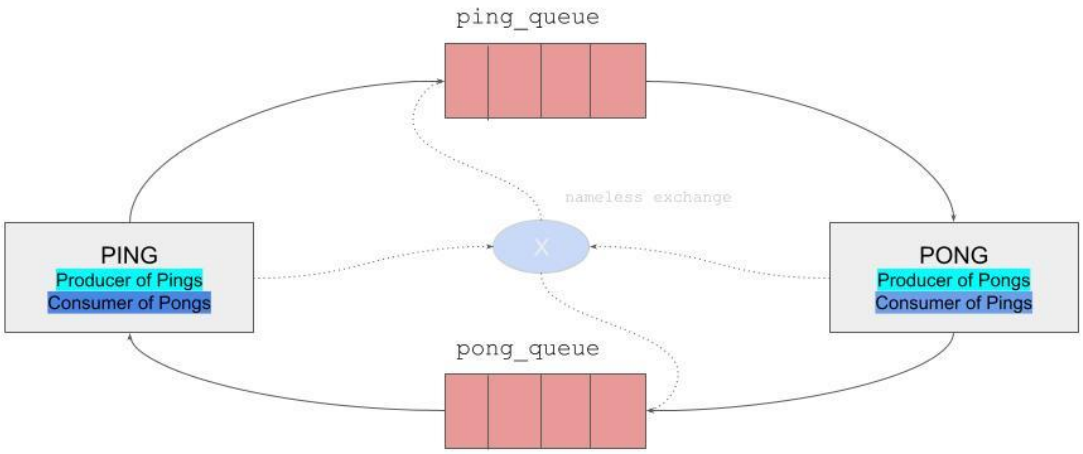

figure 1. The first basic implementation.

The ping-pong functionality is implemented in two different ways. The first one is very basic and, in a nutshell, it is a duplication of the *hello world* implementation in RabbitMQ using the default nameless direct exchange. The first ping is triggered when the ping client is run. The ping is executed by publishing to the ping queue. If a ping is received by a pong client, then the pong client publishes a pong to the pong queue which is received by the ping client. The ping client sends a new ping in response and the ping pong loop has begun. This implementation is shown in figure 1. The bindings are shown with arrows while “real” arrows to the nameless default exchange are dotted.

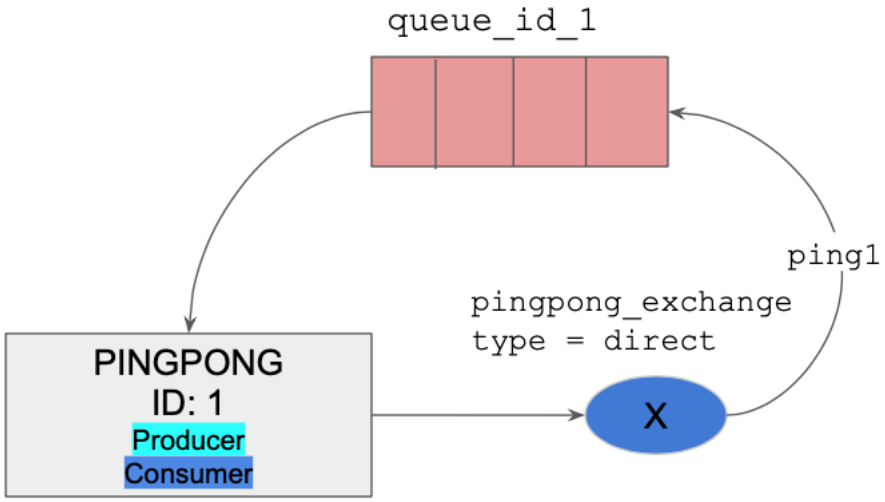

figure 2. Second implementation.

To further explore the RabbitMQ another implementation with only one type of client was done. In this implementation, when a client is started, a direct exchange is initiated. This is bound with a binding key which steers pings to the client. This is shown in figure 2.

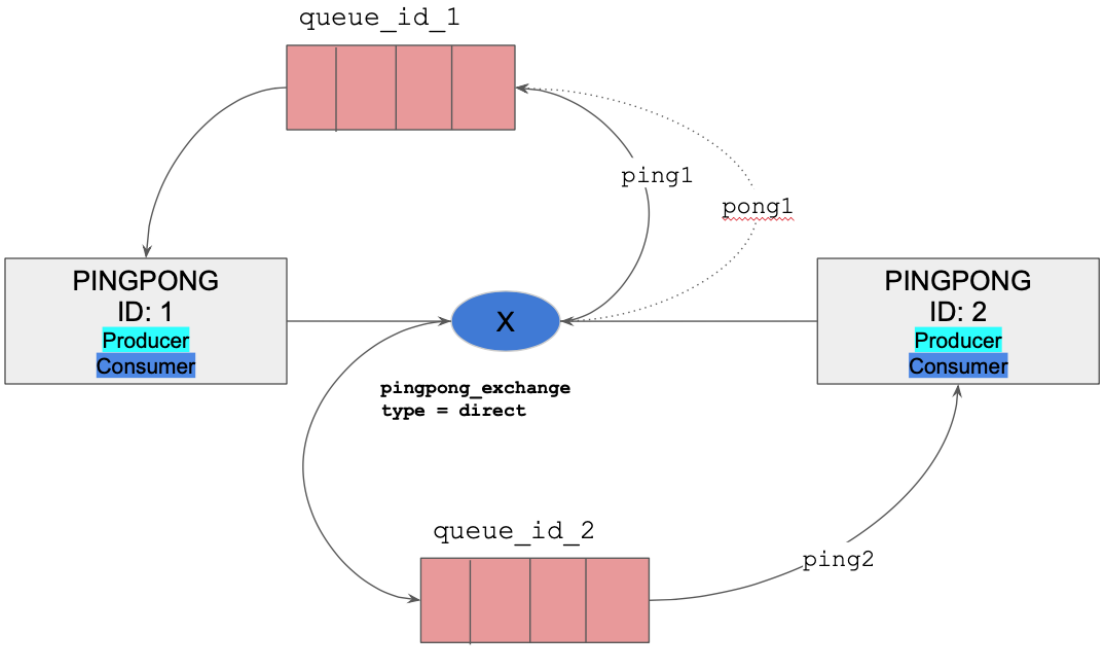

figure 3. The second implementation with two clients.

Figure 3 shows the set up when another client connects to the same exchange. A new unique queue is generated with a corresponding binding. If Pingpong 1 would ping Pingpong 2 it would remove the ping1 binding and create a pong binding with its ID. This makes it not pingable but able to receive pongs. When the other Pingpong client with ID 2 receives the ping it sends a pong with a routing key corresponding to client

one's new pong binding: pong 1. Client 1 responds and the loop has begun. The information on which client to respond to is carried in the message, for example, Pingpong 2 will receive "PING from 1".

How to run the code

The code can be found on https://github.com/Tukktuk/dist_sys-lab3
It should be run and compiled according to the instruction in the lab with the needed jar files. The Pingpong class should be run with one argument indicating the client's ID.