RabbitMQ Performance Analysis

GHITA Bogdan, SCPD
TOMA Claudiu-Mihai, SCPD

1. Architecture

The architecture consists of 3 types of entities: producers, consumers, rabbit nodes. The rabbit nodes will run in docker containers and together will form a rabbit cluster. Consumers and producers will be rabbit clients which will publish and consume messages from the rabbit queues.

2. Constraints

Our target for the tests will be the rabbit nodes. Thus, we need to make sure that the bottleneck will be on the rabbit cluster[1] and not on other parts of the system (consumers, producers, network bandwidth).

3. Metrics

The main metrics that reflect the performance of RabbitMQ are:

- Message Delivery Rate
- Message Publish Rate

Secondary performance indicators can be:

- CPU (idle, user, system, iowait)
- Memory (free, cached, buffered)
- Disk I/O (reads & writes per unit time, I/O wait percentages)
- Free Disk Space
- Network throughput (bytes received, bytes sent) vs. maximum network link throughput
- System load average (/proc/loadavg)

4. Workflow

0. define test cases
1. deploy architecture (producers, consumers, rabbit nodes, monitoring tools)
2. create scripts that will run test cases:
   - generate messages
   - manage consumers/producers/rabbit nodes (start/stop)
   - start/stop monitoring tools
3. generate statistics based on the output of the monitoring tools

5. Test cases

Each test case will be a variation of the values for the following parameters:

- publish rate (number of messages published per second)
- message size
- number of producers/consumers
- number of rabbit nodes in cluster
- number of queues

6. Resources

[1] https://www.rabbitmq.com/clustering.html
[2] https://www.rabbitmq.com/monitoring.html

7. Annexes

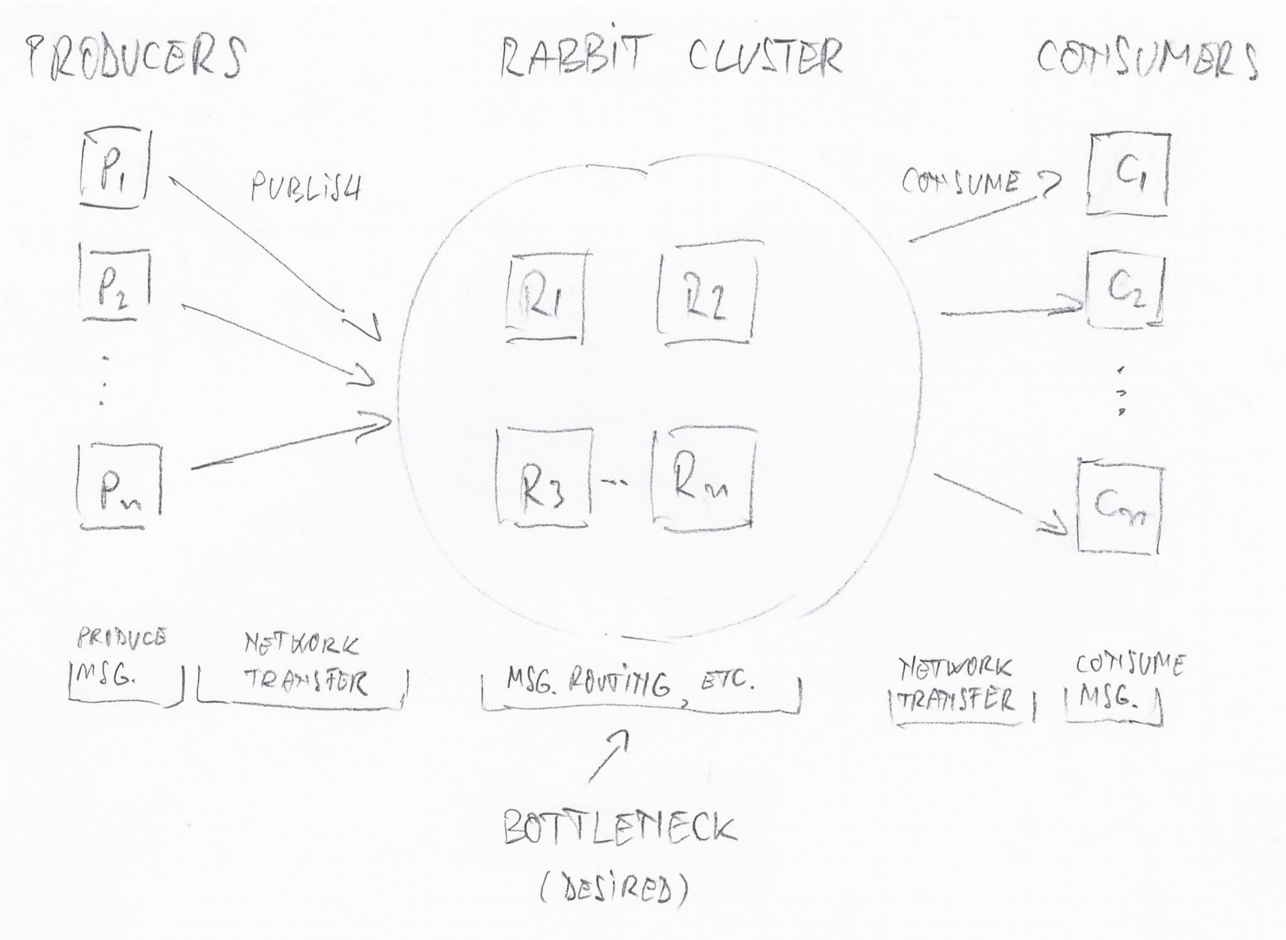

Fig. 1 Architecture