

RabbitMQ vs Kafka

RabbitMQ uses message acknowledgments to ensure delivery state on the broker itself. Kafka doesn't have message acknowledgments and it expects the consumer to remember about the delivery state.

Both of them use ZooKeeper to maintain their state across a cluster.

RabbitMQ was not designed for large volume and would fall over if the consumers were too slow.

However, post 2.0, RabbitMQ claims to handle slow batch consumers as well.

Kafka was designed from the beginning to handle both online and batch consumers.

So it can handle 100k+ events per second.

RabbitMQ's claim to that attribute is around 20k+ events per second.

Kafka uses Topic-like exchanges only.

RabbitMQ uses a richer variety of exchanges like Queues as well as Topics.

Kafka provides message ordering inside partitions.

So for strict ordering across partitions, the Kafka consumers have to be smart enough and resolve ordering across partitions themselves.

Kafka persists messages on disk and replicates them within the cluster to prevent data loss.

Each broker can handle terabytes of messages without performance impact.

Kafka has been tested to provide close to 200k messages/sec for writes and 3M messages/sec for reads.

Advanced Message Queuing Protocol (AMQP)

The Advanced Message Queuing Protocol (AMQP) is an open standard application layer protocol for message-oriented middleware.

The defining features of AMQP are:

1. Message orientation
2. Queuing
3. Routing (including point-to-point and publish-and-subscribe)

4. Reliability and
5. Security

It was originated in 2003 at JPMorgan Chase.

AMQP mandates the behavior of the messaging provider and client to the extent that implementations from different vendors are inter-operable, in the same way as SMTP, HTTP, FTP, etc. have created inter-operable systems. Previous standardizations like focused on standardizing the API level. AMQP however is a wire-level protocol. It is a description of the data-format that is sent across the network as a stream of octets. Thus it allows implementations to have different programming interfaces as long as they conform to the wire-level format.

References:

1. [Quora discussion](#)
2. [Apache Kafka](#)
3. [Advanced Messaging Queuing Protocol \(AMQP\)](#)
4. [Message brokers comparison](#)