1. Kafka has been designed originally by LinkedIn
2. Apache Kafka is a distributed publish-subscribe messaging system, which is very easy to scale out.
3. Kafka is optimized for ordered publish subscribe, while the traditional brokers are having a big feature set, which are rarely used, which affect their performance
4. Kafka provides an extremely high throughput distributed publish/subscribe messaging system.
5. Message Publishing speed is very high
6. Low Resource (CPU, memory) consumption
7. It persist messages on disk and thus can be used for batched consumption such as ETL, in addition to real time applications.

**Few reasons why Kafka output is much better are as follows:**

* Kafka producer doesn’t wait for acknowledgements from the broker and sends messages as faster as the broker can handle.
* Kafka has a more efficient storage format. On average, each message had an overhead of 9 bytes in Kafka, versus 144 bytes in ActiveMQ. This is because of overhead of heavy message header, required by JMS and overhead of maintaining various indexing structures. LinkedIn observed that one of the busiest threads in ActiveMQ spent most of its time accessing a B-Tree to maintain **message metadata and state**.
* Kafka has a more efficient storage format; fewer bytes were transferred from the broker to the consumer in Kafka.
* The broker in both ActiveMQ and RabbitMQ containers had to maintain the delivery state of every message. LinkedIn team observed that one of the ActiveMQ threads was busy writing KahaDB pages to disks during this test. In contrast, there were no disk write activities on the Kafka broker. Finally, by using the sendfile API, Kafka reduces the transmission overhead

**How to test clustering of Messaging?**

1. Configure 2 Messaging system in clustering. Create a source queue in Clustering MQ system1 and replicate the queue in Clustering MQ system2.
2. Insert a few messages on MQ system1.
3. Stop the replica,
4. Insert a few millions more messages on MQ system1.
5. Again start the replica.
6. I took only a few seconds to resync on MQ system2.

<http://www.infoq.com/articles/apache-kafka>