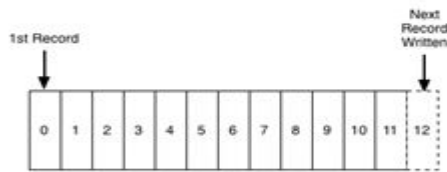
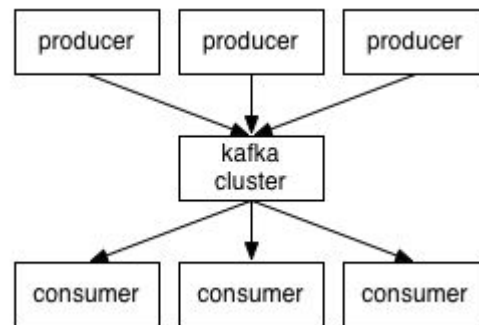


Kafka is a distributed, horizontally scalable, fault tolerant, count-log data structure.



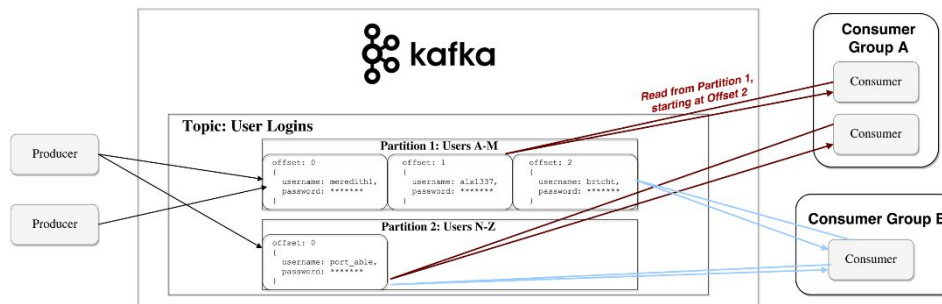
- reads and writes are $O(1)$.
- reads and writes are independent of each other, there is no locking

- Producers send messages to brokers which are stored in topics.
- Consumers are dumb, they subscribe and pool for messages.
- A broker has host several topics and topics are also replicated across multiple broker to make them fault tolerant.



Kafka Anatomy

- # Topics can get very big, it is split into several functional partitions.
- # Kafka follows dumb brokers and smart consumers ideology, message are cleared upon reaching time or memory threshold.
- # Consumers poll for messages and specify the records to read. Incrementing/decrementing the offset enables them to replay and reprocess events.
- # Consumers are actually consumer groups and each partition is tied to only consumer process per group.



Kafka persistence

- # Message grouping reduces n/w overhead, disk read/writes by batch persistence and fetching.
- # Linear read/writes on disk are fast which are further optimized by read-ahead (prefetch large block multiples) and write-behind (group small logical writes into a big physical write)

Data Distribution & Replication

- # Partition data is replicated among several brokers.
- # One broker, *Partition leader* owns the data and replicates its data to N other brokers called *followers*.
- # Zookeeper maintains the meta-data

Zookeeper

- # It is distributed key-value store used to store meta-data and mechanics of clustering.
- # It is extremely fault-tolerant.
- # consumer group's offset per partition, ACL, prod/cons quotas mes/sec, partition leaders and their health.