1. Sending data in transactions

When ur application is producing messages to kafka then use transactions in kafka by setting unique transactional id for each and every producer

1. producer batching

If u are producing large volumes of data, u can use producer batching

Means we will send data one by one, but IO threads will wait for linger.ms (some 400/X ms) or for some size, once that is met then io threads will transfer all data at once to broker

batch.size

The producer will attempt to batch records together into fewer requests whenever multiple records are being sent to the same partition. This helps performance on both the client and the server. This configuration controls the default batch size in bytes.

A small batch size will make batching less common and may reduce throughput (a batch size of zero will disable batching entirely). A very large batch size may use memory a bit more wastefully as we will always allocate a buffer of the specified batch size in anticipation of additional records

Linger.ms

This setting accomplishes this by adding a small amount of artificial delay—that is, rather than immediately sending out a record, the producer will wait for up to the given delay to allow other records to be sent so that the sends can be batched together.

Lets say u gave linger.ms=5s , means eventhough this line producer.send(new producerRecord()) ran , producer will not send this record immediately to broker

Producer will keep it in buffer for linger.ms and then IO threads will send data to broker as a batch of records, this setting is to send the data in batches

1. If u are producing large volumes of data, then u can use multi-threaded producer
2. Attaching callbacks- while sending data attach a callback, so that u will know message is sent successfully or not

producer.send(record,callback);

1. Handling duplicate data scenario- Assume duplicates will come / consider the case of same message is being retried and then solve the problem
2. Send data in Avro format – avro is very fast as less data will be transferred as schema will not be sent along with data unlike json

When same message will be retried to re consumed again and again?

1. After consuming the message, when consumer is sending back ack to broker that ack is lost due to network fluctuations, when broker didn’t received ack he will resend the same message again to consumer
2. Sometimes when consumer won’t send back ack wantedly when no follower is in sync and when only leader consumer alone consuming those messages, when followers are completely out of sync in this case also consumer wont send back ack to broker

Ex:- in sdp appn we uniquely identified a message using kafka header, we implemented database unique constraint on a column, so if same comes again since constraint is there db will unique constraint exception