

Handling Produce and Fetch Requests in KafkaApi

KafkaApis.handleFetchRequest():

- 1 parse the fetch request and make sure it is valid
- 2 if fetch request is from a follower (replicaId = -1), call maybeUpdatePartitionHW to see if the HW of the leader of a partition can be updated.
- 3 check to see if the fetch request can be satisfied immediately.
 - 3.1 If yes, call readMessageSets() immediately and send the response back.
 - 3.2 Otherwise, register a DelayedFetch watcher on each of the partition in the fetch request. The watcher will be checked and be potentially satisfied (DelayedFetch.checkSatisfied) if (a) the amount of available bytes for fetch exceeds a certain amount, or (b) a timeout has been reached.
 - 3.2.1 Condition (a) will be checked at the end of handleProducerRequest(). For each satisfied fetch request, it calls readMessageSets() to fetch the data and to send the response.
 - 3.2.2 When condition (b) is met, DelayedFetch.expire() will be called. It calls readMessageSets() to fetch whatever data is available and to send the response.

(TODO: need logic to unblock produce requests waiting on ack from followers)

KafkaApis.handleProducerRequest():

- 1 produce(): for each partition
 - 1.1 make sure leader is on this broker
 - 1.2 add new data to local log (leader replica)
 - 1.3 send response immediately (TODO: need logic to send response asynchronously when data is received by followers)
- 2 check if the produce request can satisfy an fetch request (meeting the minimum byte requirement)
- 3 for all satisfied produce requests, fetch data and send response