KIP-472: [STREAMS] Add partition time field to RecordContext

Status

Current state: *Discarded (Change is excessively complex for a non-critical issue)*

Discussion thread: *here*

JIRA: *KAFKA-7994*

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

Currently, a task in Kafka Streams has its "partition time" determined by the latest timestamp of the records it has received. This could introduce problems because when records pass through a processor whose function is to filter certain records out, some messages with the largest timestamps might be removed, allowing only records with smaller timestamps to proceed to the next task. Consequently, this means that some tasks would have the incorrect partition time.

For instance, if four records (their timestamp in the parantheses) r1(2), r2(3), r3(7), and r4(9) are making their way down the topologies, but r3 and r4 was removed, the downstream tasks would incorrectly assume that the timestamp was 3 (since last received record was r2) when in reality it should be 9 (timestamp of r4). This requires us to propagate the correct stream time through the DAG of tasks, which Kafka Streams currently does not do. However, the only manner in which an upstream and a downstream task could communicate is through the repartition topic they share. This means that any metadata related to partition time must be stored in the ProducerRecord that is sent from SinkNode to SourceNode. Currently, any modification to records, including the addition of headers or extra metadata, is fundamentally still a public API change, hence why this KIP is necessary.

Public Interfaces

The core problem that we wish to solve here is how to transmit the relevant partition time data in between tasks (via Producer/Consumer records). The following is the description of what is currently stored in a ProducerRecord:

- Key and value (serialized into a byte array)
- Topic, partition, and timestamp (A String, a 32-bit Integer, and a 64-bit Integer respectively)
- Headers (used to store extra metadata which the user wishes to pass in between tasks)

From the list above, the headers are derived from the metadata stored in RecordContext. No other parameter used in the construction of a ProducerRecord may be modified to prevent deleterious results. This in effect leaves us with two options. Modify ProducerRecord so that it can store partition time, or append a header (which stores the partition time) to the ones that are to be inserted into the ProducerRecord. If we wish to take this approach, we would add the following header to the preexisting headers that existed in RecordContext: `new RecordHeader("partition-time", ByteBuffer.allocate(Long.BYTES).putLong(partitionTime))` where `partitionTime` is a long value that stores the local partition timestamp. Therefore, what this KIP proposes is to add a new header to the ones already in RecordContext and use them in the creation of ProducerRecord.

This new header addition serves a dual purpose: it would help resolve a recurring issue found in Kafka Streams as described above, and at the same time, provides the user with more information on the current state of processing partitions (i.e. the partition time of that particular partition).

Compatibility, Deprecation, and Migration Plan

The migration plan would be decided after we have ascertained the approach we will take.

Rejected Alternatives

We will choose to not modify Producer/Consumer record as the scope of the change would not be necessary.