KIP-715: Expose Committed offset in streams

This page is meant as a template for writing a KIP. To create a KIP choose Tools->Copy on this page and modify with your content and replace the heading with the next KIP number and a description of your issue. Replace anything in italics with your own description.

Status

Current state: *"Accepted"*

Discussion thread: *here [Change the link from the KIP proposal email archive to your own email thread]*

JIRA: *here and here*

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

Motivation

If you are a Kafka user and you want to track the progress a client is making you can use the returned committed offset. However Streams does not have that capability as several Kafka clients are embedded in Streams client. This KIP proposes to add methods to TaskMetadata that report the progress Kafka exposes already. Any health check service will just need to query all clients and aggregate their responses to get a complete picture of all tasks.

Public Interfaces

TaskMetadata.java

```
package org.apache.kafka.streams.processor;

/**
* This function will return a map of TopicPartitions and the highest committed offset seen so far
*/
public Map<TopicPartition, Long> committedOffsets();

/**
* This function will return a map of TopicPartitions and the highest offset seen so far
*/
public Map<TopicPartition, Long> endOffsets();

/**
* This function will return the time task idling started, if the task is not currently idling it will return
empty
*/
public Optional<Long> timeCurrentIdlingStarted();
```

Proposed Changes

In order to make these more accessible we will allow localThreadMetadata() to be called in all states, not just running or rebalancing. This is not a change but the localThreadMetadata() returns the ThreadMetadata for each thread. That ThreadMetadata includes the TaskMetadata for each Task in the thread, this allows for the added methods to be used. There is the potential to have a task missing if it is during a rebalance and the task is being reassigned.

committedOffsets

This method will return a map that contains a mapping of all TopicPartitions and their committed offsets. The committed offset will be the highest seen so far.

endOffsets

Similarly this method will return a map that contains a mapping of all TopicPartitions and their high watermark offset.

timeCurrentIdlingStarted

This will record the time the task started idling. Once the task has stopped idling it will return empty. This also lets the user verify if the task is idling properly as they can compare the time with the config for the max idling time.

Compatibility, Deprecation, and Migration Plan

Since methods are only added and no other methods are modified, this KIP should not

- affect backward-compatibility
- deprecate public interfaces
- need a migration plan other than adding the new metrics to its own monitoring component

Rejected Alternatives

- *reporting these as metrics*
  - *We chose not to do this because we should mirror the consumer. It uses committed() method and doesn't report these with metrics*
- *adding a committed offset method to kafka streams*
  - *This turns out to be unnecessary as we can just add the information to TaskMetadata which is already exposed. Now we can avoid complicating KafkaStreams further.*