KIP-545: support automated consumer offset sync across clusters in MM 2.0

Status

Current state: Adopted

Discussion thread: https://lists.apache.org/thread.html/4ed8071a77cd8ddf7a7fe1feb4473134b2e7272a45ec180a23a6ab1e@%3Cdev.kafka.apache.org%3E

JIRA: ⚠ Unable to render Jira issues macro, execution error.

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

Motivation

MirrorMaker 2.0 (MM 2.0) is a new Kafka community initiative that is designed to address the limitations of current MirrorMaker. MM 2.0 is based on the Kafka Connect framework and has the ability to dynamically change configurations, keep the topic properties in sync across clusters and improve performance significantly by reducing rebalances to a minimum.

MM 2.0 provides a client (MirrorClient) to translate the consumer offsets from one cluster to the ones that makes sense to another cluster. Currently this offset translation can only be driven by some external tooling that will create an instance of the MirrorClient by providing cluster information in a separate config file, and run the client in a separate process by calling "remoteConsumerOffsets()" function.

The offset translation is great feature to serve the foundation of migrating or failing over downstream consumers (including Kafka stream applications) from the primary to the backup cluster, as the consumers will use the translated offsets to resume the consumption from where they left off at the primary cluster, without losing messages or consuming many duplicate messages. This expectation essentially contributes to a smooth and transparent one-time migration of consumers from one to another cluster, or the failover of consumers from primary to backup cluster.

To achieve the above transition, there are two important steps: (1) consumer offsets can be translated into the ones that make sense in another cluster, which is already done by the current MM 2.0. (2) periodically synchronize the translated offsets to the *__consumer_offsets* topic, so that when the consumers switch over to the other cluster, they can start off from the last known and translated offsets.

This KIP is mostly to propose (2) as a new feature, called "automated consumer offset sync across clusters" in MM 2.0, and discuss the initial design and how the interface looks like.

Public Interfaces

The automated consumer offsets sync will be enabled by a new config in MM 2.0 configuration file, called "sync.group.offsets.enabled", together with "emit.checkpoints.enabled". Setting both to true will make the existing "MirrorCheckpointTask" to additionally sync the selected and translated consumer group offsets (e.g. that are not active in target cluster) to the target cluster, the frequency of offset sync is same as the frequency of emitting checkpoints.

By default, the automated consumer offsets sync is not enabled. Here is an example of how to enable the one-way sync from the cluster labelled "primary" to the cluster labelled "backup":

enable automated consumer offset sync

```
primary->backup.sync.group.offsets.enabled = true
```

Proposed Changes

Overall, this KIP will not change the existing behaviors and functionalities of MM 2.0.

Write translated consumer offsets to target cluster

This is the new implementation introduced by this KIP. Current MM 2.0 already provided an interface to read and translate the consumer offsets. The next thing to do is to write the translated consumer offsets to the target cluster each time when the sync task was running. Only selected consumer offsets are written and the initial criteria are (1) only write offsets for the consumers who are inactive in target cluster. This will avoid the situation when the two consumer instances (with same consumer group ID) are running both at primary and backup clusters, the offsets at target cluster will be overwritten by the sync task. (2) if the "watermark" of the consumer offsets at target cluster is higher than the offsets at primary cluster, do not write the lower 'watermark" to target cluster. This will avoid the situation when the consumption progress at primary cluster is slower than the progress at backup cluster, writing lower 'watermark" will rewind the consumer to previous offsets, leading to consuming duplicate messages.

Compatibility, Deprecation, and Migration Plan

There is no impact to current usage, as the automated consumer offsets sync is not enabled by default. When users want to use this new feature for migrating consumers or achieving automated and transparent failover, it is suggested to enable it in lower environment as the starting point.

Rejected Alternatives

None.