KIP-134: Delay initial consumer group rebalance

Status

Current state: *Accepted*

Discussion thread: *here*

JIRA: KAFKA-4925

Released: 0.11.0

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

Motivation

When a new consumer group is created the current behaviour will result it at least two rebalances for groups with more than a single member. These rebalances can become expensive depending on what needs to be initialized and torn down. For example, during a rebalance of stateful streams processing application it will typically need to initialize any local state, often by replaying from a changelog. When partitions are revoked state needs to be persisted, which can involve flushing to disk, snapshotting, and writing to a changelog.

Consumers often don’t all start up at the same time resulting in many rebalances. As each rebalance has to go through the initialization and close phases, it can become a lengthy process to get an application into a steady running state.

Adding a configurable delay to the initial rebalance of a new, or empty, consumer group, can significantly reduce the overhead associated with starting up a stateful application by reducing the number of rebalances. Resulting in less state shuffling and wasted initialization of state that is destined to be moved, potentially multiple times. This does however come at the cost of their being additional rebalance latency for groups with a single member. As this is only on the initial join we feel this is a good trade-off.

Public Interfaces

We will add a new broker config: group.initial.rebalance.delay.ms

This config will specify the time, in milliseconds, that the GroupCoordinator will use to delay the initial rebalance when the first member joins an empty group.

Proposed Changes

Add the group.initial.rebalance.delay.ms to KafkaConfig with a default value of 3 seconds.

Change the GroupCoordinator so that upon receiving the initial JoinGroupRequest for a new or empty group, it will change the group’s state to InitialRebalance and delay the operation by min(rebalanceTimeout, group.initial.rebalance.delay.ms). If new members arrive when the group is in the InitialRebalance state, they will be added to the group and the delay will be reset to min(remainingRebalanceTimeout, group.initial.rebalance.delay.ms) . When the delay expires the group’s state will transition to PreparingRebalance and the join group will be completed.

Compatibility, Deprecation, and Migration Plan

- Existing users will see a delay in the initial rebalance. If this is not desired they can configure group.initial.rebalance.delay.ms to zero.

Test Plan

- Will add system tests with various settings timeouts and ensure we get the correct number of rebalances.
- Ensure all existing System/Integration/Unit tests continue to behave as before.

Rejected Alternatives

- Add the delay config to the consumer and add new field to JoinGroupRequest: This would result in a protocol bump.
- Use session.timeout.ms as the initial delay: We felt that this overloading of session.timeout.ms would be both confusing to users and make it more difficult to configure for all scenarios.
- Derive the delay from session.timeout.ms: Same as above
- Fixed hard-coded delay: We didn’t feel that there is a magic delay that is going to work for everyone.