KIP-423: Add JoinReason to Consumer Join Group Protocol

Current state: *Under Discussion*

Discussion thread: TBD

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

The goal of this KIP continues the effort in KIP-345 to mitigate unnecessary rebalances in order to achieve better consumer performance. Today consumer group triggers rebalance on the following circumstances:

1. *A new member joins the group with UNKNOWN_MEMBER_ID*
2. *A known member joins with changed metadata*
3. *A leader rejoins the group*
4. *A current member gets session timeout/leaves the group*

We already aim to address 1, 4 through KIP-345 by applying group.instance.id to recognize members as static throughout restarts, and avoid sending LeaveGroupRequest when the member is under static membership, thus only using session timeout to kick off expired members. For circumstance 2, it is clear that if a member has metadata update such as assignment protocol change should require another rebalance to address this change as necessary. However, we have space to improve circumstance 3, because it is not a valid condition to trigger rebalance for the most time when the leader instance is just doing a restart under static membership. It is beneficial to distinguish whether leader is rejoining for the sake of rebalance, or is rejoining just due to service restart. By specifying the join reason of the request could entirely avoid rebalance during normal consumer bounces.

In addition, as we are promoting incremental rebalances such as KIP-415, eventually we will hope to support stateful consumers such as Kafka Streams group to have new member only taking in standby task and give them time to replay the state when first joined. These new followers need to indicate a change of status when they have finished replaying the state and ready to take active tasks. If no JoinReason is specified, brokers will not be able to distinguish the joiner's purpose: whether you are requiring an incremental rebalance, or you are just joining for restart?

Furthermore, the JoinReason could serve as a helpful hint when we debug consumer rebalances in the historical perspective. Broker just needs to memorize the cause of the state transition to PrepareRebalance by the group.instance.id and JoinReason to sort out a past timeline that could cross validate the delay from topic metadata change towards group leaders' notification. There is a big room to improve upon this feature that worths wider discussion.

In conclusion, having JoinReason to gracefully handle the problem of rebalance necessity could simplify the rebalance protocol iteration, reduce unintended state shuffle and hide low-level details to brokers' perspective.

Public Interfaces

We will add a new enum field to the JoinGroupRequest interface, and bump the protocol version to v6:

```
JoinGroupRequest => GroupId SessionTimeout RebalanceTimeout MemberId GroupInstanceId ProtocolType
GroupProtocols JoinReason
  GroupId             => String
  SessionTimeout      => int32
  RebalanceTimeout          => int32
  MemberId            => String
  GroupInstanceId     => String
  ProtocolType        => String
  GroupProtocols      => [Protocol MemberMetadata]
  Protocol            => String
  MemberMetadata      => bytes
  JoinReason                 => Enum // new
```

JoinGroupRequest.java

```
public static Schema[] schemaVersions() {
    return new Schema[] {JOIN_GROUP_REQUEST_V0, JOIN_GROUP_REQUEST_V1, JOIN_GROUP_REQUEST_V2,
JOIN_GROUP_REQUEST_V3, JOIN_GROUP_REQUEST_V5, JOIN_GROUP_REQUEST_V6};
}
```

JoinGroupResponse.java

```
public static Schema[] schemaVersions() {
    return new Schema[] {JOIN_GROUP_RESPONSE_V0, JOIN_GROUP_RESPONSE_V1, JOIN_GROUP_RESPONSE_V2,
JOIN_GROUP_RESPONSE_V3, JOIN_GROUP_RESPONSE_V4, JOIN_GROUP_REQUEST_V5, JOIN_GROUP_REQUEST_V6};
}
```

In the meantime, two JoinReason enums will be introduced to handle the leader rejoin case for the very first version:

JoinReason.java

```
public enum JoinReason {
  RESTART("restart"), // Join request from a restart/newly started consumer
  TOPIC_METADATA_CHANGE("topic_metadata_change"); // The topic metadata has changed (must be from the leader)
  UPGRADE("upgrade"), // The client is doing upgrade that requires rebalance.
}
```

Proposed Changes

Detecting a change of topic metadata is currently the only case when leader consumer wants to trigger a group rebalance. We will explicitly set the JoinReason to `topic_metadata_change` so that group coordinator will proceed to rebalance stage when hitting this reason. For members rejoin with UNKNOWN_MEMBER_ID. the rebalance will still trigger because the join reason is implicitly conveyed as "requiring a new member identity and grow the group" which is reasonable to trigger rebalance. For members joining with identity (either known member.id or known group.instance.id), if the JoinReason is specified as `RESTART`, stable group won't trigger rebalance since this indicates a restart happens on this member and nothing should be affected for the entire group.

Be aware that JoinReason takes lower priority than current rebalance logic checking such as protocol change or unknown member join. That's why dynamic member's behavior will not be affected.

Compatibility, Deprecation, and Migration Plan

- *This change doesn't involve backward incompatible change. Broker will still be able to read existing join group request, and for request < v6 the JoinReason field will be interpreted as "unknown" which shouldn't affect broker's judgement on whether to trigger rebalance by existing conditions.*
- *User needs to restart both the broker and consumer fleet to enable this new feature. The specific order doesn't matter.*

Rejected Alternatives

Not applicable so far.