KIP-352: Make leader aware of partition reassignment

Status

Current state: *Discussion*

Discussion thread:

JIRA:

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

Motivation

From the perspective of the leader, there is no such thing as a partition reassignment. There is only the current replica set. This has several drawbacks:

1. While a reassignment is in progress, the new replicas are trying to catch up. The broker considers these partitions under-replicated even if the desired replication factor is always satisfied. This is misleading and makes URP metrics difficult to alert on.
2. During a reassignment, users often attempt to throttle the reassignment traffic in order to avoid negatively impacting the cluster. The throttle applies to any replica which is not in sync, but there is no way for the leader to distinguish the replicas which are being reassigned. If a replica falls out of sync, then suddenly it gets hit with the throttle. Not only does this make the replica slower to catch back up, it increases the amount of traffic which is subject to the throttle which could make the reassignment go even slower.
3. As replicas are catching up, they are added to the ISR. Depending on the rate that the replicas catch up, there may be a non-trivial amount of time during which the ISR is larger than the desired replication factor. This can negatively impact end-to-end latency since replication must await all members of the ISR. Perhaps even worse, it makes determining the correct throttle even trickier because all ISR traffic skips the throttle.

This KIP proposes to make the leader aware of partition assignments.

Proposed Changes

The problem at the moment is that only the controller knows about the reassignment. Partition leaders just see a single replica set. We propose to have the controller propagate the reassignment state to the leaders. We will distinguish between the current set of replicas and the impending set of replicas. The impending replica set will contain the new replica assignment while the reassignment is in progress.

Public Interfaces

Request APIs

We will modify the UpdateMetadata and the LeaderAndIsr request APIs to allow the controller to propagate the new reassignment to the leaders. The new LeaderAndIsr request schema is given below:

```
LeaderAndIsrRequest => ControllerId ControllerEpoch [PartitionState] [LiveLeader]
  ControllerId => INT32
  ControllerEpoch => INT32
  PartitionState => TopicName PartitionId ControllerEpoch LeaderId LeaderEpoch ISR ZkVersion ActiveReplicas
ImpendingReplicas IsNew
    TopicName => STRING
    PartitionId => INT32
    ControllerEpoch => INT32
    LeaderId => INT32
    LeaderEpoch => INT32
    IsNew => BOOLEAN
    ZkVersion => INT32
    ISR => [INT32]
    CurrentReplicas => [INT32] // Renamed
    TargetReplicas => [INT32]  // New
```

Similar changes will be made to the UpdateMetadata request.

```
UpdateMetadataRequest => ControllerId ControllerEpoch [PartitionState] [LiveLeader]
  ControllerId => INT32
  ControllerEpoch => INT32
  PartitionState => TopicName PartitionId ControllerEpoch LeaderId LeaderEpoch ISR ZkVersion ActiveReplicas
ImpendingReplicas
    TopicName => STRING
    PartitionId => INT32
    ControllerEpoch => INT32
    LeaderId => INT32
    LeaderEpoch => INT32
    ZkVersion => INT32
    ISR => [INT32]
    CurrentReplicas => [INT32] // Renamed
    TargetReplicas => [INT32]  // New
```

The response schemas for both APIs will match the previous version.

Metrics

We will change the semantics of the "UnderReplicated" metric to count only the partitions which are under-replicated from the perspective of the active replica set. We will add a new metric "ReassignedCount" which tracks the number of replicas which are currently being reassigned.

Compatibility, Deprecation, and Migration Plan

The main concern from a compatibility perspective is the semantic change to the "UnderReplicated" metric. Users may have to make changes if this is used to track the reassignment state. However, we believe that continued misuse of this metric (i.e. not taking reassignment into account) is a more substantial problem.

Rejected Alternatives

We considered leaving the "UnderReplicated" metric with its current semantics and adding a new metric to represent the "under-synchronized" replicas. We ultimately rejected this because we felt it was necessary to address the misuse of the URP metric due to its surprising behavior during a reassignment.