KIP-550: Mechanism to Delete Stray Partitions on Broker

Status

Current state: Discussion

Discussion thread: here

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

Stray partitions could be left on the broker's disk in certain scenarios. Stray partitions are ones that are not known to the controller, are not present in the replica state in ZK and are not being actively used by any clients. Specifically, we could end up with stray partition(s) on a broker when when partition reassignment moves replicas off of an offline broker. If the broker is (or happens to be) offline when the reassignment completes, the controller would not get a chance to send the StopReplicaRequest. When the broker starts up, the partition will remain on-disk forever from this point on.

The broker has no mechanism to clean up such stray partitions. This becomes problematic because

1. The broker opens a Log instance for all partitions on its local disk, including stray partitions.
2. Retention would not delete any segments for stray partitions. Retention starts deleting segments only when the high watermark is higher than the segment's last offset. A stray partition is not a valid replica anymore, and thus has no defined high watermark. This means the disk space for stray partitions can never be reclaimed.
3. If a broker hosts a stray partition and the topic is recreated, there is no protection in place to be able to distinguish the current stray partition from the new partition. In the worst case, this means data for the previous generation of the partition will now reside in the current generation.

This KIP proposes a mechanism to clean up stray partitions, solving problems (1) and (2) listed above. (3) is mitigated to an extent but we would require the improvements that are part of KIP-516: Topic Identifiers.

Public Interfaces

We propose to change the LeaderAndIsrRequest to include an additional `containsAllReplicas` field, denoting whether the request contains the full replica list hosted by the target broker.

```
diff --git a/clients/src/main/resources/common/message/LeaderAndIsrRequest.json b/clients/src/main/resources
/common/message/LeaderAndIsrRequest.json
index 852968801..7ddca80b9 100644
--- a/clients/src/main/resources/common/message/LeaderAndIsrRequest.json
+++ b/clients/src/main/resources/common/message/LeaderAndIsrRequest.json
@@ -22,7 +22,9 @@
// Version 2 adds broker epoch and reorganizes the partitions by topic.
//
// Version 3 adds AddingReplicas and RemovingReplicas
- "validVersions": "0-4",
+ // Version 4 adds flexible versions
+ // Version 5 adds ContainsAllReplicas
+ "validVersions": "0-5",
"flexibleVersions": "4+",
"fields": [
{ "name": "ControllerId", "type": "int32", "versions": "0+", "entityType": "brokerId",
@@ -51,7 +53,9 @@
"about": "The leader's hostname." },
{ "name": "Port", "type": "int32", "versions": "0+",
"about": "The leader's port." }
- ]}
+ ]},
+ { "name": "ContainsAllReplicas", "type": "bool", "versions": "5+",
+ "about": "Whether the request contains all replicas hosted by the target broker." }
],
"commonStructs": [
{ "name": "LeaderAndIsrPartitionState", "versions": "0+", "fields": [
```

Proposed Changes

Today, when a new broker starts up, the controller sends a full list of replicas the broker hosts in the LeaderAndIsrRequest. We will formalize this contract by adding the `containsAllReplicas` field to the request. On a new broker startup or on controller failover, the controller will send LeaderAndIsrRequest containing the full set of replicas and will also set `containsAllReplicas` to `true`. When a broker receives a LeaderAndIsrRequest with `containsAllReplicas` set to `true`, it can safely use the replica list in this request as the source-of-truth for the partitions it must host. Any partitions the broker hosts that are not present in the LeaderAndIsrRequest will then be scheduled for deletion, as those would constitute stray partitions.

Note that we have the ability to detect outdated requests with KIP-380, so that would still apply before stray partition detection could kick in, ensuring we don't make this decision based on an outdated request sent to an old generation of the broker. The broker also ensures that the LeaderAndIsrRequest is sent by the latest controller and fences any other, fencing off requests from a zombie controller.

Compatibility, Deprecation, and Migration Plan

There is no impact on compatibility, deprecation or migration concern with this KIP.

Rejected Alternatives

NA