KIP-637: Include min.insync.replicas in MetadataResponse to make Producer smarter in partitioning events

Status

Current state: Under Discussion

Discussion thread: here

JIRA: KAFKA-10230

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

Motivation

Currently Kafka Producer is able to identify unavailable partitions and avoid routing messages to them, but the definition of an unavailable partitions is - the leader of the partition is not available, this may have covered most of the cases to avoid unnecessary network calls etc.

From Producer point of view, to successfully transmit a message may require certain number of acks to be responded from the cluster. When the acks of Producer is set to all, partition leader availability is not able to define if a partition is ready/available to accept requests, Producer also needs to know if there are enough in-sync replicas in the cluster.

By exposing the min.insync.replicas of each Kafka topic to Producer, we could potentially avoid exhausting the entire Producer buffer and unnecessary network calls when a partition does not have enough in-sync replicas in the cluster and new replication may take a while to complete (partition is too large).

Public Interfaces

MetadataResponse

Add one more property called minInSyncReplicas to the TopicMetadata class that located in MetadataResponse class.

MetadataReponse.TopicMetadata

```
public static class TopicMetadata {
    private final Errors error;
    private final String topic;
    private final boolean isInternal;
    private final List<PartitionMetadata> partitionMetadata;
    private int authorizedOperations;
    private int minInSyncReplicas;
}
```

MetadataResponseTopic.SCHEMA_10

Add one more field min_insync_replicas to the MetadataResponseTopic response and bump up the version.

MetadataResponseTopic.SCHEMA_10

```
public static final Schema SCHEMA_10 =
    new Schema(
        new Field("error_code", Type.INT16, "The topic error, or 0 if there was no error."),
        new Field("name", Type.COMPACT_STRING, "The topic name."),
        new Field("is_internal", Type.BOOLEAN, "True if the topic is internal."),
        new Field("partitions", new CompactArrayOf(MetadataResponsePartition.SCHEMA_9), "Each partition in the
topic."),
        new Field("topic_authorized_operations", Type.INT32, "32-bit bitfield to represent authorized
operations for this topic."),
        new Field("min_insync_replicas", Type.INT16, "min.insync.replicas of this topic."),
        TaggedFieldsSection.of(
        )
    );
```

Partitioner

Add one extra parameter requireMinInSyncAcks to the partition method of Partitioner, so that we are able to build a readyPartitions on top of the availablePartitions that we get from cluster.availablePartitionsForTopic(topic) and partition messages by the readyPartitions.

Partitioner

```
/**
 * Compute the partition for the given record.
 *
 * @param topic The topic name
 * @param key The key to partition on (or null if no key)
 * @param keyBytes The serialized key to partition on( or null if no key)
 * @param value The value to partition on or null
 * @param valueBytes The serialized value to partition on or null
 * @param requireMinInSyncAcks The value of acks of producer
 * @param cluster The current cluster metadata
 */
public int partition(String topic, Object key, byte[] keyBytes, Object value, byte[] valueBytes, boolean
requireMinInSyncAcks, Cluster cluster);
```

Proposed Changes

The KafkaApis.handleTopicMetadataRequest needs to be updated to populate min_insync_replicas to MetadataResponseTopic

Add one more parameter requireMinInSyncAcks to Partitioner.partition method so that we are able to build a readyPartitions on top of the availablePartitions that we get from cluster.availablePartitionsForTopic(topic) and partition messages by the readyPartitions. When the acks of producer is set to 0 or 1, readyPartitions is identical to the availablePartitions we get from *cluster.*

Compatibility, Deprecation, and Migration Plan

The new field will not be populated when new clients talk to older brokers or when old clients talk to newer brokers, so there shouldn't be any compatibility issue.

Rejected Alternatives