KIP-912: Support decreasing send's max block time without worrying about metadata's fetch

Status

Current state: *Under Discussion*

Discussion thread: *here*

JIRA: Kafka-14768

Motivation

Sometimes, application's threads will block for *max.block.ms* to send records using KafkaProducer#send. It exhausted threads of whole system for the time in some cases.

When application try to reduce the *max.block.ms* to decrease the blocking time. They will find they couldn't change the value to any one which is smaller than the time costed for metadata's fetch. What's more, metadata's fetch is one heavy operation which cost a lot of time.

Take our project as example. we will take about 4 seconds to complete the metadata's fetch. So, we can't change the *max.block.ms* to any value < 4000ms.

After analyzing the issue. The root cause is the configured *max.block.ms* is shared by "metadata fetch" operation and "append record" operation. We can refer to follow table in detail:

where to block	**when it is blocked**	**how long it will be blocked?**
org.apache.kafka.clients.producer. KafkaProducer#waitOnMetadata	The first request which need to load the metadata from kafka	<max.block.ms
org.apache.kafka.clients.producer.internals. RecordAccumulator#append	At peak time for business, if the network can't send message in short time.	<max.block.ms

What's more, the metadata's fetch only need to be done one time in KafkaProducer#send. After the complete of first fetch, the metadata will be retrieved from cache directly and its timer update only happen on network thread instead of user's thread.

So, this KIP try to reach the goal we can reduce the blocking time by changing the max.block.ms to wanted smaller value without worrying about the metadata's fetch.

Public Interfaces

No public interface changed. Just change the inner implement of private method:

org.apache.kafka.clients.producer.KafkaProducer#doSend

Add two new configure items for producer.

Proposed Changes

The changes can refer to the example PR: https://github.com/apache/kafka/pull/13335/files

Add two configures with tiny code changes related which control the timeout in KafkaProducer#send

1. Two configures added

Producer's configure.	configure item.	default value
includeWaitTimeOnMetadataInMaxBlockTime	max.block.ms.include.metadata	false
maxWaitTimeMsOnMetadata	max.block.metadata.ms	60 seconds

2. Code changes

By default, includeWaitTimeOnMetadataInMaxBlockTime is true, all of the behaviors are not changed.

When user set includeWaitTimeOnMetadataInMaxBlockTime to false, KafkaProducer#send will block maxWaitTimeMsOnMetadata for metadata's fetch and block max.block.ms for remaining operations.

Compatibility, Deprecation, and Migration Plan

If user want to use the feature, user can upgrade the client with the new configures set.

If user don't have requirement for it, there isn't any need to do any change. What's more, new client version's upgrade also won't influence existed behavior.

- *What impact (if any) will there be on existing users?*
  no impact on existed users.
- *If we are changing behavior how will we phase out the older behavior?*
  no changing older behavior.
- *If we need special migration tools, describe them here.*
  no.
- *When will we remove the existing behavior?*
  no need to remove.

Test Plan

We can test with test matrix:

if we need N (2<N<5) seconds for metadata's fetch, we will send record to test producer with different configures.

Cases to send record. \Configures	*max.block.ms*	includeWaitTimeOnMetadataInMaxBlockTime(*max.block.ms.include.metadata*)	maxWaitTimeMsOnMetadata(*max.block.metadata.ms*)
1	10 seconds	default value: false	default value: 60 seconds
2	1 seconds	default value: false	default value: 60 seconds
3	10 seconds	true	default value: 60 seconds
4	1 seconds	true	5 seconds
5	1 seconds	true	1 seconds

Case 2 and case 5 will fail to send records. All of others are success.

Rejected Alternatives

Alternative 1:

Provide new method to complete the metadata fetch not controlled by *max.block.ms* and user should call it before sending any record. For example, user can call it before marking the service ready.

The alternatives can solve the issue and also solve the first record's slow latency issue. but if user only have interesting to reduce the blocking time without care about it. It isn't the best solution for this requirement due to user may forget it to call it before any sending or don't aware to call another method to solve the issue.

Alternative 2:

Refer to https://cwiki.apache.org/confluence/display/KAFKA/KIP-286. The KIP's goal is that "We will change the behavior of producer.send() so that it does not block on metadata update".

I think the metadata's blocking is still must-have one. the thing we can do is to move the blocking before producer.send(). Thus, the KIP don't solve the issue from this point.