KIP-953: partition method to be overloaded to accept headers as well.

Status

Current state: *"Under Discussion"*

Discussion thread: https://lists.apache.org/thread/0f20kvfqkmhdqrwcb8vqgqn80szcrcdd

JIRA: *KAFKA-15187*

Motivation

Current partition method only accepts key and value. Wanted to provide support for selecting partition based on headers too. This lets users pass values to partitioner method without tampering the key and value.

Few of the use cases where this can be helpful :

1. *The user has a header which provides priority of the message [lets say 0 to 9]. You have 10 partitions which you want to use for each priority. With headers available in partition method the partition can be chosen based on the header value.*
    a. *Currently, there is an implementation which achieve the functionality by appending the header value to the key and stripping it off in the keySerializer when creating the keyBytes. This is a hacky solution.*
2. The user wants to have a decorator partitioner which they want to use to select from a different set of partitioners based some params but do not want to tamper with the key and value. The user can pass the params in headers, evaluate it in the headers and call partitioner of interest.

Public Interfaces

Partitioner interface in org/apache/kafka/clients/producer will have the following update :

- A new overloaded method partition(String topic, Object key, byte[] keyBytes, Object value, byte[] valueBytes, Cluster cluster, Headers headers) with default implementation to call the existing method.
- The object passed as headers is readonly and cannot be updated.
- Details :

```
 /**
* Compute the partition for the given record. Not overriding this method in the Partitioner interface
has the same behaviour as using the existing method. The object passed as headers is readonly and cannot
be updated.
*
* @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 cluster The current cluster metadata
* @param headers The headers to partition on or null
*/
default int partition(String topic, Object key, byte[] keyBytes, Object value, byte[] valueBytes,
Cluster cluster, Headers headers) {
return partition(topic, key, keyBytes, value, valueBytes, cluster);
};
```

Proposed Changes

- Add a new overloaded method partition(String topic, Object key, byte[] keyBytes, Object value, byte[] valueBytes, Cluster cluster, Headers headers) with default implementation to call the existing method in the Partitioner interface in org/apache/kafka/clients/producer.

    - This lets the users to implement the partitioner with ability to choose partition based on header values.
    - If a partitioner implements both the methods, the override of new method (one which accepts headers) will be in effect.
- Make the call to partition method in KafkaProducer to use the new method.
    - Since it is defaulted to use the existing method, this should not cause any behaviour change.

Total changes : https://github.com/apache/kafka/pull/13981

Compatibility, Deprecation, and Migration Plan

- *What impact (if any) will there be on existing users?*
    - *An implementation of the new interface will be incompatible with old clients.*
- *If we are changing behavior how will we phase out the older behavior?*
    - *NA*
- *If we need special migration tools, describe them here.*
    - *NA*
- *When will we remove the existing behavior?*
    - *NA*

This is a backward compatible change as mentioned in the proposed changes section. The new default method calls the existing method unless it is manually overridden by the user.

Test Plan

This has an addition of a default method in the interface. The changes made in Kafka Producer should be covered in the existing tests and should not have any impact.

Rejected Alternatives

None