KIP-22 - Expose a Partitioner interface in the new producer

Status

Current state: *Accepted*

Discussion thread: *here*

JIRA: KAFKA-2091

Released: 0.8.3

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

Motivation

In the new producer you can pass in a key or hard code the partition as part of ProducerRecord. Internally we are using a class **org.apache.kafka.producer.internals.Partitioner** . This class uses the specified partition if there is one; uses a hash of the key if there isn't a partition but there is a key; and simply chooses a partition round robin if there is neither a partition nor a key.

However there are several partitioning strategies that could be useful that we don't support out of the box. An example would be having each producer periodically choose a random partition. This tends to be the most efficient since all data goes to one server and uses the fewest TCP connections, however it only produces good load balancing if there are many producers. Of course a user can do this now by just setting the partition manually, but that is a bit inconvenient if you need to do that across a bunch of apps since each will need to remember to set the partition every time.

The idea would be to expose a configuration to set the partitioner implementation like **partitioner.class=org.apache.kafka.producer.DefaultPartitioner** . This would default to the existing partitioner implementation.

Public Interfaces

```
package org.apache.kafka.clients.producer;
public interface Partitioner extends Configurable {

/**
 * Configure partitioner class with given key,value pairs from partitioner.metadata config.
 */
 @override
 public void configure(Map<String, ?> configs);

/**
 * 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
 * @param valueBytes The serialized value to partition on
 * @param cluster The current cluster metadata
 */
 public int partition(String topic, Object key, byte[] key, Object value, byte[] value, Cluster cluster);

 /**
  * This is called when partitioner is closed.
  */
  public void close();


 }
```

```
package org.apache.kafka.clients.producer;
public class DefaultPartitioner implements Partitioner {
}
```

Proposed Changes

1. Add a new partitioner interface .
2. Move existing partitioner code to DefaultPartitioner .
3. If Users provide partition id than it will take precedence over partitioner.class .If partition id is null than only partitioner.class partition() method called to compute partition for given Record.
4. Introduce new producer config called "partitioner.class" . By default "partitioner.class" points to org.apache.kafka.clients.producer. DefaultPartitioner

Compatibility, Deprecation, and Migration Plan

- *It will not impact any of existing clients. Even when they upgrade to new version, they can continue to use their existing client code without the need to add partitioner.class config.*

Rejected Alternatives

None.