KIP-262: Metadata should include number of state stores for task

- Status
- Motivation
- Public Interfaces
- Proposed Changes
- Compatibility, Deprecation, and Migration Plan
- Rejected Alternatives

Status

Current state: *Rejected*

Discussion thread: *here*

JIRA: *KAFKA-6583*

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

Motivation

Currently, in `StreamPartitionAssignor`, assignment takes place without regards to which tasks are stateful and which are not. Thus potentially creating a situation where we have an unbalanced assignment of tasks. For `StreamPartitionAssignor` to factor in stateful tasks, the current rebalance protocol's metadata should be upgraded so that it contains the number of state stores that exists for a certain task. In this manner, when metadata is decoded in `StreamPartitionAssignor#assign(Cluster metadata, Map<String, Subscription> subscription)`, the data for stateful tasks are now available for more even distibution to clients.

Public Interfaces

The class in which the rebalance protocol's metadata is contained is `SubscriptionInfo` which encodes information into a `ByteBuffer`. What this upgrade would require is to change the format of the encoding and decoding to the `ByteBuffer` in `SubscriptionInfo`, which would require us to give an upgrade strategy for older Kafka versions. A new encoding and decoding format will be introduced: mainly the addition of an extra integer for each `Task Id` in `SubscriptionInfo`.

Proposed Changes

The following changes will be made to `TaskId` to facilitate the recording of the number of state stores for a specific task:

TaskId.java

```
public class TaskId implements Comparable<TaskId> {

    /** The ID of the topic group. */
    public final int topicGroupId;
    /** The ID of the partition. */
    public final int partition;
+   /** The number of State Stores in the task. */
+   private int numberOfStateStores;

    public TaskId(int topicGroupId, int partition) {
        this.topicGroupId = topicGroupId;
        this.partition = partition;
+       this.setNumberOfStateStores(0);
    }

        //...
        public void writeTo(ByteBuffer buf) {
        buf.putInt(topicGroupId);
        buf.putInt(partition);
+               buf.putInf(numberOfStateStores);
    }
    public static TaskId readFrom(ByteBuffer buf) {
-               return new TaskId(buf.getInt(), buf.getInt());
+       final TaskId result = new TaskId(buf.getInt(), buf.getInt());
+               result.setNumberOfStateStores(buf.getInt());
+               return result;
    }

+        public int numberOfStateStores() {
+       return numberOfStateStores;
+   }

+   public void setNumberOfStateStores(int numberOfStateStores) {
+       this.numberOfStateStores = numberOfStateStores;
+   }
}
```

In this manner, when `TaskId#encode()` and `TaskId#decode()` is called in `SubscriptionInfo`, the number of state stores will also be processed. Note that `setNumberOfStateStores(...)` will also be called during `StreamTask` instantion to set the number of state stores to the given `ProcessorTopology`'s state store list size.

It would also be helpful to know that in the older version of encoding `TaskId`, only the topic group id and partition number was included. In this newer version, the number of state stores is also made available.

Compatibility, Deprecation, and Migration Plan

Since we are updating metadata, this will undoubtedly necessitate an upgrade path which has been provided in KIP-268.

Rejected Alternatives