KIP-251: Allow timestamp manipulation in Processor API

Status

Current state: *Accepted* [VOTE] KIP-251: Allow timestamp manipulation in Processor API

Discussion thread: *[DISCUSS] KIP-251: Allow timestamp manipulation in Processor API*

JIRA: ⚠ Unable to render Jira issues macro, execution error.

Released: 2.0

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

Motivation

Kafka Streams has a defined "contract" about timestamp propagation at the Processor API level: all processors within a sub-topology, see the timestamp from the input topic record that is currently processed and this timestamp will be used for all result records when writing them to a topic, too. For punctuation, the used output record timestamp is the current event-time or wall-clock time (depending on the punctuation type) that triggers the punctuation (i.e., for event-time the timestamp of last processed record that fires the punctuation).

For the DSL and also for custom operators, it would be desirable to allow timestamp manipulation at Processor API level for individual records that are forwarded. This allows to support a larger scope of possible semantics for stream processing.

Public Interfaces

Using Processor API, users use `ProcessorContext` to forward record to downstream operators by calling `forward()`. Thus, we need to add overloads for `forward()` to allow user to pass in a timestamp for the output record.

```
package org.apache.kafka.streams.processor;

public interface ProcessorContext {
  // existing overloads of forward() that we keep
  <K, V> void forward(K key, V value);

  // existing overloads of forward() that we deprecate
  <K, V> void forward(K key, V value, int childIndex);
  <K, V> void forward(K key, V value, String childName);

  // new overloads of forward()
  <K, V> void forward(K key, V value, To to);

  // other existing methods omitted for brevity
}
```

We also add a new auxiliary class to specify optional argument for `ProcessorContext#forward()`

```
package org.apache.kafka.stream.processor;

public class To {
  private To(String childName, int childIndex, long timestamp);

  public static To child(String childName);
  public static To all();

  public To withTimestamp(long timestamp);
}
```

Proposed Changes

We add one new overload of `ProcessorContext#forward()` that take one additional parameter of type `To`. to specify optional argument like childName or output record timestamp. If users call the new overloads and set a timestamp on the `To` object, the output record gets the specified timestamp assigned. For the existing methods or if no timestamp is specified on `To`, the default contract using the input record timestamp for the output record will be used. Note, that the new `To` class does not include forwarding by index anymore—we want to remove forwarding by index and only support forwarding to all or by name.

Compatibility, Deprecation, and Migration Plan

This change is backward compatible, as we don't alter the contract of existing methods, and only add new method with new functionality.

Test Plan

We can test this feature with unit tests, by implementing test Processors that manipulate the output record timestamp using the new API, and by verifying that the output record have the assigned timestamps. Existing test ensure, that the old behavior is preserved.

Rejected Alternatives

None.