KIP-765: Introduce new SlidingWindow type for [start,end] time

Status

Current state: *Under Discussion*

Discussion thread: *here [Change the link from the KIP proposal email archive to your own email thread]*

JIRA: KAFKA-12839

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

Motivation

In KIP-450: Sliding Window Aggregations in the DSL, we introduce Sliding Window to do calculations for each *distinct* window, the sliding window implementation would be a more efficient way to do these types of aggregations. In the Sliding Window, we was planning to have windows with inclusive on both the start and end time, but currently, we implemented in TimeWindow which is a half-open time interval. We should introduce a new window type: **SlidingWindow**, to have "\[start, end]" time interval, for SlidingWindows aggregation.

Public Interfaces

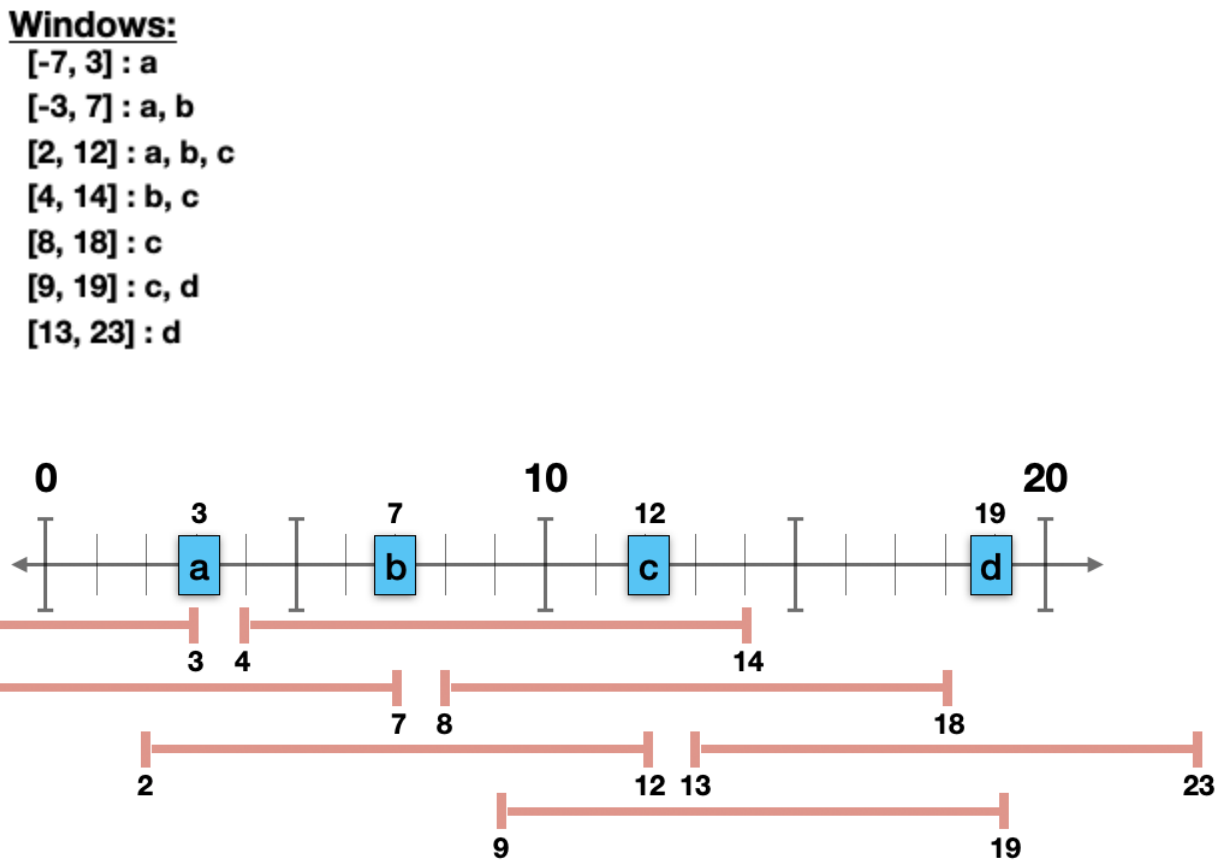
Add org.apache.kafka.streams.kstream.internals.*SlidingWindow*

```
/**
 * A {@link SlidingWindow} covers a closed time interval with its start timestamp and end timestamp as an
inclusive boundary.
 * It is a fixed size window, i.e., all instances (of a single {@link org.apache.kafka.streams.kstream.
TimeWindows
 * window specification}) will have the same size.
 * <p>
 * For time semantics, see {@link org.apache.kafka.streams.processor.TimestampExtractor TimestampExtractor}.
 *
 * @see SessionWindow
 * @see UnlimitedWindow
 * @see org.apache.kafka.streams.kstream.TimeWindows
 * @see org.apache.kafka.streams.processor.TimestampExtractor
 */
public class SlidingWindow extends Window {

    /**
     * Create a new window for the given start time (inclusive) and end time (inclusive).
     *
     * @param startMs the start timestamp of the window (inclusive)
     * @param endMs   the end timestamp of the window (inclusive)
     * @throws IllegalArgumentException if {@code startMs} is negative or if {@code endMs} is smaller than {@cod
e startMs}
     */
    public SlidingWindow(final long startMs, final long endMs) throws IllegalArgumentException { }

    /**
     * Check if the given window overlaps with this window.
     *
     * @param other another window
     * @return {@code true} if {@code other} overlaps with this window&mdash;{@code false} otherwise
     * @throws IllegalArgumentException if the {@code other} window has a different type than {@code this}
window
     */
    @Override
    public boolean overlap(final Window other) throws IllegalArgumentException {}
}
```

Proposed Changes

This KIP will change the KStreamSlidingWindowAggregate implementation, to replace the original TimeWindow type, with the new **SlidingWindow** type. So that we can have correct aggregation calculation for the set of records within the window.

Ex: In KIP-450, we have a graph to describe what expected result we want:



But because of we used the TimeWindow to represent the window time interval, it actually has 1 ms difference for each window.

ex: originally, we expected have a window \[-7, 3], but we actually have a window \[-7, 2], or \[-7, 3)

After this KIP, this issue will be fixed and have a correct aggregation calculation results.

Compatibility, Deprecation, and Migration Plan

No migration plan is needed.

Rejected Alternatives