KIP-362: Support dynamic gap session window

Status

Current state: *"Under Discussion"*

Discussion thread: *here*

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

Motivation

Currently, Kafka Streams DSL only supports fixed-gap session window. However, in some circumstances, the gap is more dynamic and can vary depending on other factors: the statistical aggregation result, liquidity of the records, etc. In such cases, allowing the user to define a dynamic-gap session is important. This KIP proposes a way to add such support into Kafka Streams DSL, by allowing extract dynamic gap from each record. Through this enhancement, a user can calculate the gap in upstream processors of the topology, then "windowedBy" using DynamicSessionWindow.

Public Interfaces

We are proposing following changes to public interfaces:

Add a ""windowedBy" method to KGroupedStream

```
SessionWindowedKStream<K, V> windowedBy(final DynamicSessionWindows windows);
```

Define a new session window type DynamicSessionWindows

```
public final class DynamicSessionWindows {
private SessionWindowGapExtractor gapExtractor;
private final long maintainDurationMs;
private final Duration grace;

DynamicSessionWindows(final SessionWindowGapExtractor gapExtractor, final long maintainDurationMs, final
Duration grace) {
  this.gapExtractor = gapExtractor;
  this.maintainDurationMs = maintainDurationMs;
  this.grace = grace;
}

  public static DynamicSessionWindows withDynamicGap(final SessionWindowGapExtractor gapExtractor) {
    if (gapExtractor == null) {
     throw new IllegalArgumentException("Gap extractor (gapExtractor) cannot be null.");
    }
    final long oneDayMs = 24 * 60 * 60_000L;
    return new DynamicSessionWindows(gapExtractor, oneDayMs, null);
  }

  ...
}
```

A new extractor interface to extract gap from record

```
public interface SessionWindowGapExtractor {
    long extract(Object k, Object v);
}
```

Proposed Changes

Four new classes are added:

DynamicSessionWindow: similar to existing SessionWindow, except it includes a SessionWindowGapExtractor instead of a fixed gap value.

SessionWindowGapExtractor: an interface with extract() method, to extract gap value from a record.

KStreamDynamicSessionWindowAggregate: implement necessary interfaces for dynamic session window processor.

DynamicSessionWindowedKStreamImpl: implement SessionWindowedKStream interface.

A new overloaded method windowedBy is added into KGroupedStream. It takes a DynamicSessionWindow as parameter.

Grace period and retention period

There are some recent changes proposed on KIP-328 – specifically here – which are related to this change. To align with that:

The grace period for dynamic session window is user-configurable, similar to the manner in which it is handled in fixed-gap session window.

On the other hand, the retention period needs to be handled differently. For fixed-gap session window, the retention period is set as **max(user-configured-retention, fixed-gap)**, with a default value as 24*60*1000 (1 day). Since the gap won't change in this case, the state store can be built using this pre-computed retention period.

For dynamic-gap session window, the retention period can no longer be pre-computed using existing methodology. The user should explicitly set the retention period for dynamic-gap window. For any gap extracted from the record, if it is larger than the configured retention period, the gap will be automatically adjusted to the retention period. If it is not configured, the same default 1 day will be used.

Compatibility, Deprecation, and Migration Plan

- *should be no compatibility issue*

Rejected Alternatives

None