

KIP-261: Add Single Value Fetch in Window Stores

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Rejected Alternatives](#)

Status

Current state: *Adopted (2.0.0)*

Discussion thread: *TBD*

JIRA:



Unable to render Jira issues macro, execution error.

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

Motivation

For windowed state stores in Kafka Streams, today we only provide range query APIs to search through a time range (or unbounded range) for a single key (or a range of keys).

However, if users know which window they are querying exactly, they should be able to issue a "single-point" query just like the key-value store within that window.

And the implementation of such a single point query would be much less costly than a range query.

For example, for windowed aggregations in Streams DSL, the underlying implementation is leveraging the `fetch(key, from, to)` API to get all the related windows for a single record to update. However, this is a very inefficient operation with significant amount of CPU time iterating over window stores (preliminary experiment results with RocksDB shows 10X difference). On the other hand, since the operator implementation itself have full knowledge of the window specs it can actually translate this operation into multiple single-point queries with the accurate window start timestamp, which would largely reduce the overhead.

Public Interfaces

This KIP would propose to add a single fetch API to the `WindowStore`, and use that in the `KStreamWindowAggregate` / `KStreamWindowReduce` operators.

`ReadOnlyWindowStore` interface

```
V fetch(K key, long windowStartTimestamp);
```

Where the `windowStartTimestamp` parameter is used as the unique identifier of specified time window.

Proposed Changes

1. For all implementations of the `WindowedStore`, including `CachingWindowStore`, `RocksDBWindowStore`, etc, add the implementations of this API for a single value lookup accordingly.
2. In `KStreamWindowedAggregate` and `KStreamWindowedReduce`, replace the `"fetch(K, long start, long to)"` with this newly added API.

Compatibility, Deprecation, and Migration Plan

- Users implementing their own state stores would be affected by the interface changes. More specifically:
 - For user's customized window store implementation, they need to make code changes to implement this additional API. And then recompile their code before doing the rolling bounce to upgrade to the new version.

Rejected Alternatives

- We have also considered adding this as an internal interface and do the optimization only for DSL implementations to avoid breaking code. We think this is not worthy as we don't expect people to have lots of custom window store implementations and thus only a very small fraction of users (if any) would be affected.