

KIP-385: Avoid throwing away prefetched data

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

Status

Current state: *Under Discussion*

Discussion thread: [here](#)

JIRA: [here](#)

Motivation

Kafka consumer pipelines the fetching of data in order to maximise performance. Whenever `poll(Duration)/poll(long)` is called before any results is returned, another fetch is issued. Albeit benefitting performance, in some circumstances when combined with the use of the `pause/resume` API, this optimisation can result in transferring quite a bit of duplicate data over the wire. The reason for this to happen is that whenever `poll` is called any prefetched data is thrown away in case the topic-partition is paused. To illustrate the effect with a simple example, imagine that a single `KafkaConsumer` instance is assigned two topic partitions `TP1` and `TP2`. Since the client interested in `TP1` cannot handle records as fast than the one in `TP2`, we resort to pausing `TP1` whenever we are not interested in receiving records for it. This results in the following behavior:

1. `TP1` is resumed and `poll` is called on it, where `poll` returns some data
2. The consumer issues a fetch request in order to pre-fetch the next batch of records for `TP1`
3. `TP2` is resumed and `TP1` paused (as the consumer of `TP1` is not ready for more records)
4. All prefetched records for `TP1` are now thrown away.
5. This cycle repeats indefinitely

This KIP proposes an improvement that allows us to use instead of throwing away the prefetched data.

Public Interfaces

No change

Proposed Changes

The change proposed here is simply to not throw the prefetched data and keep it around until the partition is resumed and polled again.

Compatibility, Deprecation, and Migration Plan

No breaking changes.

Rejected Alternatives

None