KIP-349: Priorities for Source Topics

Status

Current state: *Under Vote*

Discussion thread:

https://lists.apache.org/list.html?dev@kafka.apache.org:lte=1M:kip-349

Must use Advanced search options to go back in history as discussion has not been active since October 2018.

JIRA: *KAFKA-6690*

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

Motivation

To support use-cases where there's a need to prioritize source topics. If a consumer is subscribed to more than one topic, i.e., a high priority topic *High* and a low priority topic *Low*. The consumer should consume events from topic *Low* only when all the events from topic *High* are consumed.

From the reporter of KAFKA-6690:

We use Kafka to process the asynchronous events of our Document Management System such as preview generation, indexing for search etc.
The traffic gets generated via Web and Desktop Sync application. In such cases, we had to prioritize the traffic from web and consume them first.

Public Interfaces

The addition of new subscribe API that allows caller to prioritize topics. New class `TopicPriority` constructor

```
public TopicPriority(java.lang.String topic, int priority);
```

where `priority` is a positive integer.

This `subscribe` method takes a list of `TopicPriority` as a parameter. Starvation of lower priority topics is not addressed in this KIP and a possible consequence of invoking the API.

```
public void subscribe(java.util.List<TopicPriority> topicPriorities);
```

Proposed Changes

The new behavior is only in effect when the user specifies priorities for topics using the above new API. In this case topics are checked starting from highest priority first and then in descending order based on priority. All events must be consumed from a higher priority topic before consumption is performed on a lower priority topic.

Note that the issue of starvation of lower priority topics has been discussed and is intended by design.

Compatibility, Deprecation, and Migration Plan

This feature preserves backward compatibility by treating topics without priorities as having the same priority (i.e. current consumer semantics does not change).

Rejected Alternatives

None