KIP-1003: Signal next segment when remote fetching

This page is meant as a template for writing a KIP. To create a KIP choose Tools->Copy on this page and modify with your content and replace the heading with the next KIP number and a description of your issue. Replace anything in italics with your own description.

Status

Current state: "In Discussion"

Discussion thread: https://lists.apache.org/thread/1s1l97yxwwwr93vwx378n68v7pmjb5k5

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

In Tiered Storage, RSM implementations tend to pre-fetch data from remote tier (e.g. S3) to optimize for sequentially reading (from oldest to newest data) until eventually local data is reached.

Currently, this pre-fetching technique is limited to the current segment being fetched:

As there is no reference/state on the RSM implementation about the ordering of segments — unless it stores the metadata in advanced, though this should be discouraged as the metadata is already managed by the RLMM.

Fetching performance can be optimized so much as the size of the segment allows, causing inconsistent latency as fetch requests go across segments.

To improve this, this KIP proposes to include a new parameter on the `RemoteStorageManager#fetchLogSegment` operation to signal the next segment when fetching. This way, depending on how far on the current segment the consumer is reading, it can decide to start preparing data from the next segment.

This should allow RSM implementations to support *pre-fetching across segments* and have a more predictable performance when reading from remote tier.

Public Interfaces

The following methods are proposed to add to the `RemoteStorageManager` interface:

```
    default InputStream fetchLogSegment(RemoteLogSegmentMetadata remoteLogSegmentMetadata,
                                        Optional<RemoteLogSegmentMetadata> nextRemoteLogSegmentMetadata,
                                        int startPosition) throws RemoteStorageException {
        return fetchLogSegment(remoteLogSegmentMetadata, startPosition);
    }

    default InputStream fetchLogSegment(RemoteLogSegmentMetadata remoteLogSegmentMetadata,
                                        Optional<RemoteLogSegmentMetadata> nextRemoteLogSegmentMetadata,
                                        int startPosition,
                                        int endPosition) throws RemoteStorageException {
        return fetchLogSegment(remoteLogSegmentMetadata, startPosition, endPosition);
    }
```

Proposed Changes

From the Tiered Storage framework, the RemoteLogManager can reuse the existing logic to fetch the next remote segment metadata and pass it to the RSM implementation.

Compatibility, Deprecation, and Migration Plan

To be backward compatible, a default implementation mimicking the current behavior is provided.

Test Plan

The new methods will have to be implemented and tested by the RSM implementations.

RemoteLogManager tests will be extended to cover the scenarios when there is and there is not a next segment metadata available.

Tiered Storage integration tests will be extended to cover the new APIs and test it similarly to the existing ones.

Rejected Alternatives

If there are alternative ways of accomplishing the same thing, what were they? The purpose of this section is to motivate why the design is the way it is and not some other way.