KIP-47 - Add timestamp-based log deletion policy

Status

Current state:*Under Discussion*

Discussion thread: *here*

JIRA: *KAFKA-3224*

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

Motivation

One of Kafka's officially-described use cases is a distributed commit log (http://kafka.apache.org/documentation.html#uses_commitlog). In this case, for a distributed service that needed a commit log, there would be a topic with a single partition to guarantee log order. This service would use the commit log to re-sync failed nodes. Kafka is generally an excellent fit for such a system, but it does not expose an adequate mechanism for log cleanup in such a case. With a distributed commit log, data can **only** be deleted when the client application determines that it is no longer needed; this creates completely arbitrary ranges of time and size for messages, which the existing cleanup mechanisms can't handle smoothly.

A new addition to the existing deletion policy based on the absolute timestamp of a message would work perfectly for this case. The client application will periodically update the minimum timestamp of messages to retain, and Kafka will delete all messages earlier than that timestamp using the existing log deletion mechanism, alongside the existing size-based and duration-based checks.

This is based off of work done for KIP-32 - Add timestamps to Kafka message. and KIP-33 - Add a time based log index.

Public Interfaces

This KIP has the following public interface changes:

- Expose a new topic configuration, log.retention.min.timestamp. The value will be a Unix time in milliseconds.

Proposed Changes

- Add a new topic configuration, **log.retention.min.timestamp.**
  - The format of the value will be a Unix time in milliseconds.
- Modify the log deletion mechanism (in LogManager.scala) to also delete segments whose last timestamp is before the configured timestamp if the timestamp is set
  - This check will use the time-based log index from KIP-33 - Add a time based log index.
- Timestamp-based deletion will work with both **CreateTime** and **LogAppendTime** timestamp types.

Compatibility, Deprecation, and Migration Plan

There are no backwards compatibility or migration concerns with this change. The default value of the timestamp will be zero, so it will be ignored unless explicitly configured.

Rejected Alternatives

All of these alternatives are meant to make it simple for client applications to delete a specific range of messages from a given partition on demand.

- Pluggable log compaction policy. This would allow users to build a custom predicate to apply to log messages during compaction, and deploy it alongside brokers. This approach is more complex and harder to test.
- Admin tool to truncate a particular partition up to a minimum offset. Spiked implementation of this and it became fairly complex because of the need to track the completion of a given deletion command on each broker in Zookeeper, to be able to delete Zookeeper metadata created by these commands.
- Manually setting / resetting existing configuration settings to try to flush ranges of a given partition's log. The various ways one would do this are not exact, and rely on several asynchronous processes that make coordinating it very tricky.