KIP-468: Avoid decompression of record when validate record at server in the scene of inPlaceAssignment .

Status

Current state: *Under Discussion*

Discussion thread: *here [Change the link from the KIP proposal email archive to your own email thread]*

JIRA: *KAFKA-8364 [Change the link from KAFKA-1 to your own ticket]*

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

Motivation

We do performance testing about Kafka server in specific scenarios .We build a kafka cluster with one broker,and create topics with different number of partitions.Then we start lots of producer processes to send large amounts of messages to one of the topics at one testing .And we found that when the upper limit of CPU usage has been reached But it does not reach the upper limit of the bandwidth of the server network(**Network inflow rate:600M/s; CPU(%):>97%**).

We analysis the JFIR of Kafka server when doing performance testing .After we checked and completed the performance test again, we located the code " **ByteBuffer recordBuffer = ByteBuffer.allocate(sizeOfBodyInBytes);**(**Class:DefaultRecord,Function:readFrom()**)" which consumed CPU resources and caused a lot of GC .So we remove the allocation and copying of ByteBuffer at our modified code, the test performance is greatly improved(**Network inflow rate:1GB/s;CPU(%):<60%**)

We also analysis the code of validation to record at server. **Currently the broker will decompress whole record including 'key' and 'value' to validate record timestamp, key, offset, uncompressed size bytes, and magic . We remove the decompression operation** and then do performance testing again . we found the **CPU's stable usage is below 30% even lower. Removing decompression operation to record can minimize CPU usage and improve performance greatly.**

Should we think of preventing decompress record when validate record at server in the scene of inPlaceAssignment ?

We think we should optimize the process of server-side validation record for achieving the purpose of verifying the message without decompressing the message. Maybe we can add some properties ('batch.min.timestamp'(Long) , 'records.number'(Integer),'all.key.is.null'(boolean)) in client side to the batch level for validation, so that we don't need decompress record for validate 'offset','timestamp' and key (The value of 'all.key.is.null' will false when there is w key is null).

Public Interfaces

Code class change maybe include:

We can add the '**batch.min.timestamp'(Long**) , '**records.number'(Integer)**,'**all.key.is.null'(boolean)** to schema of RecordBatch . Those properties will be used to validate record . Of course the magic value > 1.

- **RecordBatch**
- **DefaultRecordBatch**
- **LogValidator**

Proposed Changes

We want to optimize the process of server-side validation record without decompressing the message, so that it can minimize CPU usage and improve performance.

Compatibility, Deprecation, and Migration Plan

- Such a change will not affect the previous magic version(Magic value <=1) to do validate record.

Rejected Alternatives

None