KIP-824: Allowing dumping segmentlogs limiting the batches in the output

Status

Current state: *Adopted*

Discussion thread: here

Voting thread: here

JIRA: here

Motivation

Basically this change will allow us to save plenty of money on detecting poor configuration of the producers and monitoring them in near-real time.

As a devops engineer maintaining a big infrastructure of Kafka clusters I want to have visibility on the way the clients (producers) produce the data.

The way the batches are produced directly affects the cluster resources and performance, so we can say producers and brokers resources are tightly coupled.

As it is known batching and compressing brings a huge saving in terms of resources, worth to mention the payload content is important as well.

In general our clients are not aware of the way they batch and compress and we don't know either as we don't have a way to monitor this. I started using the kafka-dump-log.sh script which is great for inspecting the produced batches.

Unfortunately, the script does not use the **slice** method of FileRecords object which could partially read a part of the segment log file instead of the whole file.

Reading the whole file(s) drastically reduce the usage of this script as this potentially affect a production environment when reading several files in a short period of time, and at the end just reading a few MB or batches will give us the needed information of the current pattern on the topic.

Ideally it would be nice to have a way to take just a sample of the segment log without reading the whole file in memory to avoid removing the useful page cache that Kafka is using.

In an ideal world producers send the data the OS will keep the producer data in disk and memory as well if the consumers are reading near by the end then in general they consume from “memory” .

On the other hand if we run a script to inspect all the topics to know if the clients are compressing and batching we start reading thousands of files and several GB of data which could start breaking the armony and then consumers will need to “get” the data from the disk.

Having a way to just read a sample of the segment logs without reading the whole files will allow us to monitor in “real time” sending metrics to our monitoring solution.

Also my idea is to create a script which will use the kafka-dump-log.sh periodically, detect missing compression and batching, take some samples (including the payloads) and simulate a compression.

Public Interfaces

- As it is shown below the **kafka-dump-log.sh** script will support a new parameter called **--max-bytes** which will limit the amount of batches read from the log segment.
- When the parameter is not set the script will work as usual, there is not any breaking change.

Executing command

```
$ bin/kafka-dump-log.sh --max-bytes 5000 --files /var/lib/kafka/data/test-topic-0/00000000013997524812.log
Dumping /var/lib/kafka/data/test-topic-0/00000000013997524812.log
Starting offset: 13997524812
baseOffset: 13997524812 lastOffset: 13997524819 count: 8 baseSequence: 0 lastSequence: 7 producerId: -1
producerEpoch: 0 partitionLeaderEpoch: 219 isTransactional: false isControl: false position: 0 CreateTime:
1646237831158 size: 1100 magic: 2 compresscodec: GZIP crc: 68255305 isvalid: true
baseOffset: 13997524820 lastOffset: 13997524822 count: 3 baseSequence: 0 lastSequence: 2 producerId: -1
producerEpoch: 0 partitionLeaderEpoch: 219 isTransactional: false isControl: false position: 1100 CreateTime:
1646237832108 size: 930 magic: 2 compresscodec: GZIP crc: 38868740 isvalid: true
baseOffset: 13997524823 lastOffset: 13997524824 count: 2 baseSequence: 0 lastSequence: 1 producerId: -1
producerEpoch: 0 partitionLeaderEpoch: 219 isTransactional: false isControl: false position: 2030 CreateTime:
1646237832174 size: 763 magic: 2 compresscodec: GZIP crc: 2646429942 isvalid: true
baseOffset: 13997524825 lastOffset: 13997524836 count: 12 baseSequence: 0 lastSequence: 11 producerId: -1
producerEpoch: 0 partitionLeaderEpoch: 219 isTransactional: false isControl: false position: 2793 CreateTime:
1646237832258 size: 1478 magic: 2 compresscodec: GZIP crc: 1905889818 isvalid: true
```

- Parameter description: **"Limit the amount of total batches in bytes avoiding reading the whole file(s)."**

Proposed Changes

The **kafka-dump-log.sh** uses the **DumpLogSegments.scala** which uses the object FileRecords for reading the content of the segment log.

I added the **slice** method which supports passing the amount of bytes to be read (**end**) parameter.

Then the end the batch iterator will return the InputStream only reading the amount of bytes passed as parameter instead of the whole file

As I mentioned above the change requires to call the slice method in FileRecords class to allow to pass the **end** parameter.

FileRecords open

```
   val fileRecords = FileRecords.open(file, false).slice(0, maxBytes)
```

- The code changes can be seen here in the open PR

Rejected alternatives

- It was mentioned to fix the parameter "max-messsage-size" (a different feature) as it is not working properly, it was discarded due to the potential confusion with the current implementation, see more here.
- Use number of batches instead of byes, you can see why this was discarded here

Compatibility, Deprecation, and Migration Plan

- There is not any impact on existing users, it is adding a new feature via an optional parameter,
- I am using Integer.MAX_VALUE as a default value for **maxBytes** as FileRecords accepts Integer for **end** parameter, that means we previously had a limitation of this value. So when the new parameter is not passed it will send the integer.MAX_VALUE which will be the equivalent of reading the whole file (if it does not exceeds the 2 GB) right now based on FileRecord class this limitation already exists