KIP-964: Have visibility when produce requests become "async"

- Status
- Motivation
- Proposed Changes
- Compatibility, Deprecation, and Migration Plan
- Test Plan
- Rejected Alternatives

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: *Under Discussion*

Discussion thread: *here*

JIRA: If the idea is approved then I will create the Jira

Motivation

The main motivation is to have a clear metric (in spite of the OS) to see when the produce requests become "async" .In a normal situation the produce requests will be written to disk via teh lib->syscall->etc.., as we know this will end up in a memory page (dirty page from now on)

If the system is not under pressure all the writes (produce requests, compactions) are async as they will be written to disk when we reach the background threshold of the OS to write the dirty pages. In the other hand if the dirty pages reach the "hard" limit then all the writes will become "sync" as they have to wait until the OS flushes the dirty pages.

This has a huge impact on the produce requests, we are talking about nanoseconds to milliseconds (writing in memory vs disk). From the producer side this generates timeouts and other errors as the produce requests get to the leader, go to the purgatory to wait for the acks (if set) and finally the producer gets the OK.

We had this problem in production as we had to replace AWS EBS volumes in order to shrink them, as we know in AWS you need to add a new volume and copy all the data. Also we throttle the data stream from the other brokers but that is out of the scope.

the new volumes streamed the data from the other brokers generating a lot of new data to be written to disk, this + compacitons + produce requests reached the hard limit of the dirty pages forcing the OS to start the sync writes and drastically degrading the produce requests.

As far as the new broker is copying partitions they become again the leaders and start getting produce requests, but the broker is still under pressure because of the data being streamed,

In our case we were able to address this issue playing a bit with the OS resources and the OS dirty pages configs, but it would have been great if we had a metric to monitor when the produce requests get close to become "sync"

With that monitor we can be proactive while scaling horizontal/vertical the cluster instead of doing this when the problem is already in place

Basically an indicator to see when our broker is "sync" or "async" for the produce requests

I checked Confluent already has a metric:

`kafka.log:`*`type=SegmentStats,name=SegmentAppendTimeMs`* ***The time in milliseconds to append a record to the log segment.***

This metric is the perfect to give visibility this issue, the only downside is when you see this the problem is already in place.

For a perfect metric we have to go to OS checks which is out of the scope

Of course this is a still good metric as at least will allow us to have a quick understanding of the bottleneck when produce request times spikes and start generating issues in the producer side

Public Interfaces

- *Monitoring*

Proposed Changes

Describe the new thing you want to do in appropriate detail. This may be fairly extensive and have large subsections of its own. Or it may be a few sentences. Use judgement based on the scope of the change.

We can track the metric here something like this:

```
object SegmentAppendStats {
  private val metricsGroup = new KafkaMetricsGroup(SegmentAppendStats.getClass)
  val SegmentAppendTimer: Timer = metricsGroup.newTimer("SegmentAppendRateAndTimeMs", TimeUnit.MILLISECONDS,
TimeUnit.SECONDS)
}
```

Compatibility, Deprecation, and Migration Plan

- I need confirmation if tracking this metric could have a performance impact (Thanks in advance)

Test Plan

If the KIP is accepted I can easily test the scenario producing records, checking the new metric before and after (sync vs async) writes

I can play using the dirty_ratio and background_dirty_radio values.

Rejected Alternatives

The best alternative IMHO would be to get the information before "the disaster happens" so at OS level we can check the ***nr_dirty*** *and the* ***nr_dirty_threshold***

nr_dirty is the amount of current dirty pages and nr_dirty_threshold is the limit when the OS will block the writes in the pages until some are flushed.

Having this relation could give us a hint when we are getting closer to the limit and add more resources or tune the OS settings.

This is possible as an "in house" metric but not for Kafka as it runs in the JVM and only god know in which OS 🙂