

An analysis of the impact of max.in.flight.requests.per.connection and acks on Producer performance

This page is a summary of results on the analysis I did on understanding the optimal value of max.inflight.requests.per.connection as well as the performance impact of acks=all.

Test Setup

- 3 brokers on AWS, d2.xlarge instances: 3x2TB locally attached disks. 32GB RAM, 4 Xeon cores
- 1 client machine in same availability zone.
- Each performance run produced 10GB of data.

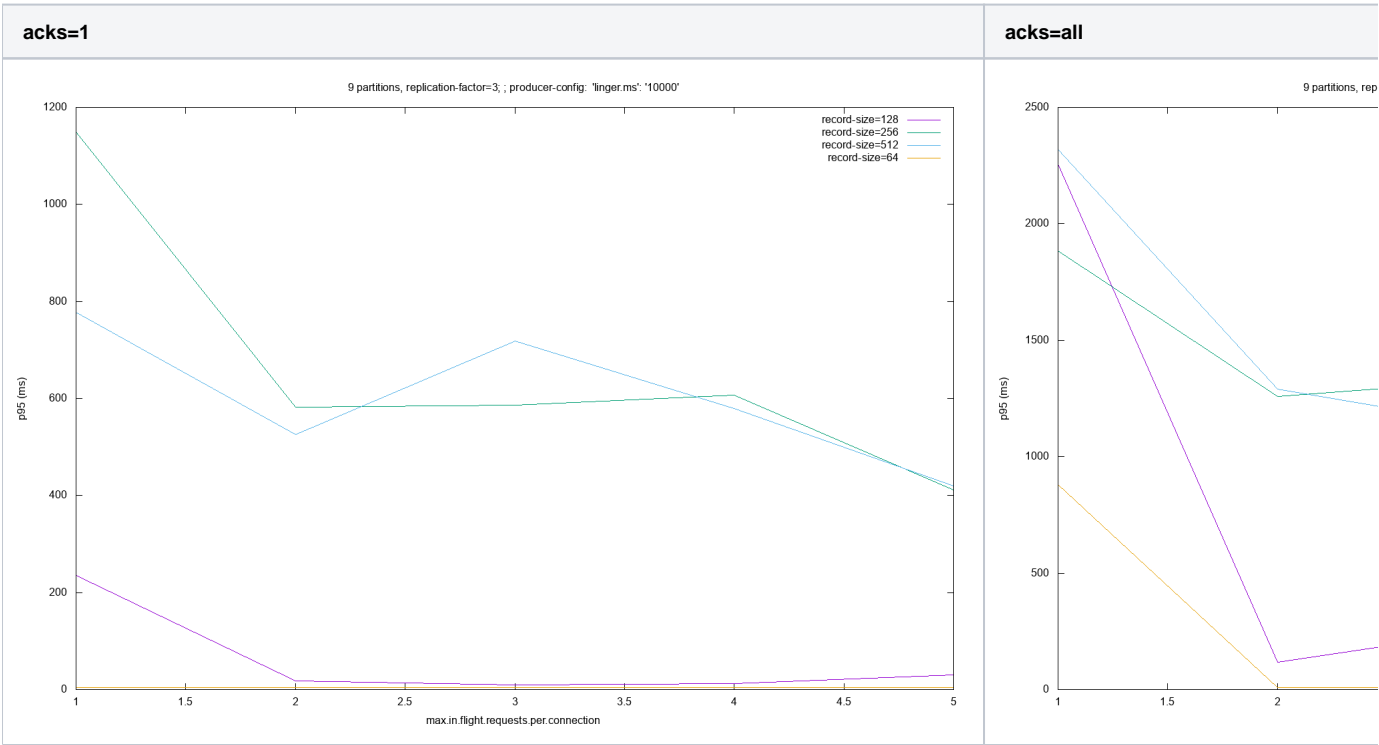
Tests run against kafka commit [6bd73026](#).

Goal

- Understand the performance curve for different values of max.in.flight.requests.per.connection.
 - We expect better throughput and latency for higher values of this variable. But when do the benefits tail off?
 - If we want to support max.inflight > 1 when enabling idempotence, should we pick a single value and not allow further configuration? If so, what should this value be?
- Understand the effect of acks=all when compared to acks=1. If it is slower why? Can we make acks=all the default?

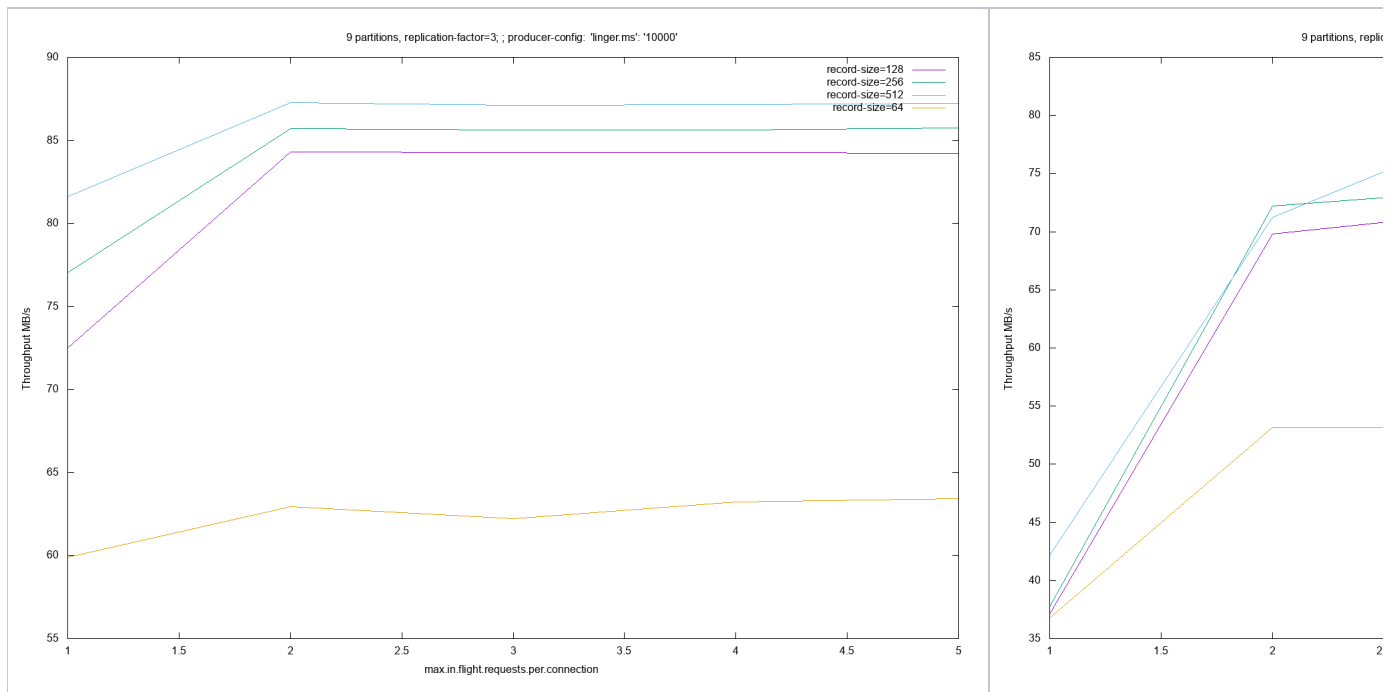
Summary of results

p95 Latency



Throughput

| acks=1 | acks=all |
|--------|----------|
|--------|----------|



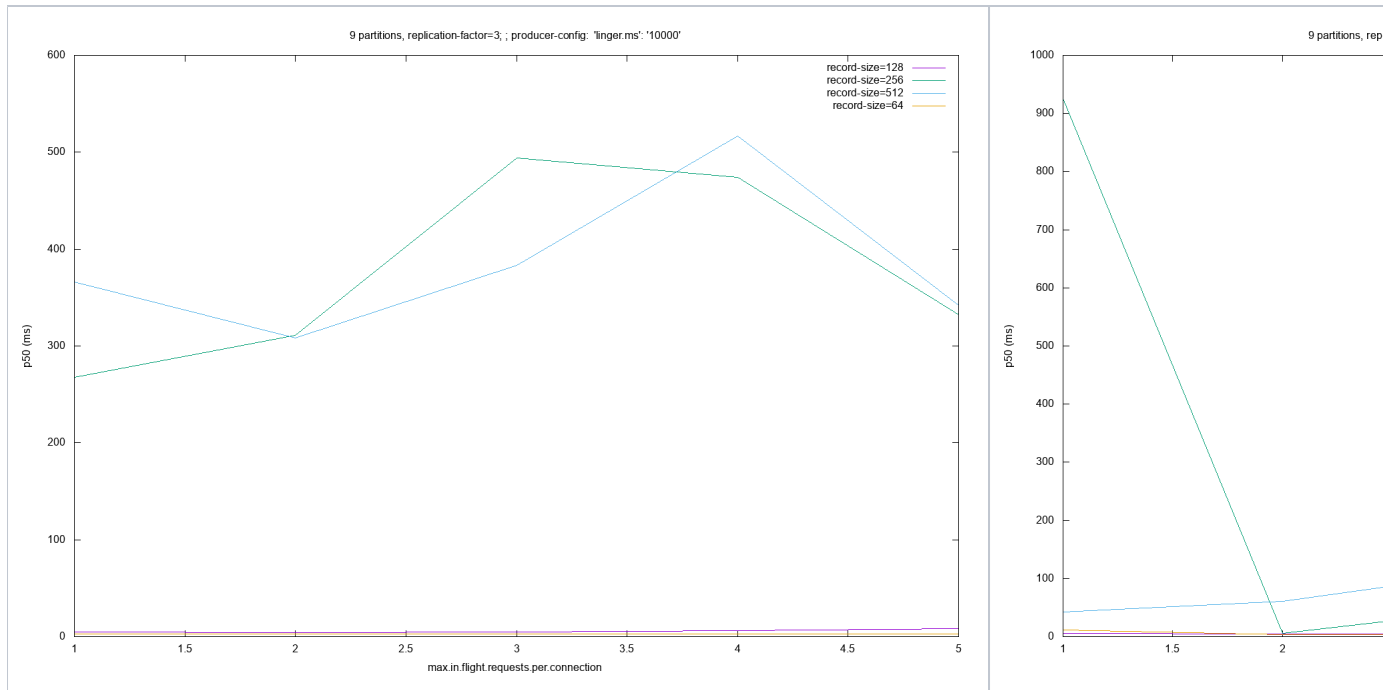
Observations

- Throughput and latency show big improvements from max.inflight=1 to max.inflight=2, but the performance plateaus thereafter.
- Slight throughput degradation between acks=1 and acks=all.
- There is a major 2x degradation in p95 latency between acks=1 and acks=all except for 64 byte messages.
- Plots above are for 9 partitions. If you keep increasing the number of partitions, the difference between acks=1 and acks=all and max.inflight=1 and max.inflight=2 becomes smaller and smaller.
 - This not surprising as as the number of partitions increases, the payload of each `ProduceRequest` is bigger, hence the relative overhead of additional operations per request is smaller.

More on acks=1 and acks=all

For the run above, the p50 latency for acks=1 and acks=all is totally unintuitive.. it is actually better for acks=all, and also is worse for max.inflight=4 when compared to max.inflight=3

| acks=1 | acks=all |
|--------|----------|
|--------|----------|



At this time, there is nothing to explain the performance behavior of acks=all and acks=1:

- Broker metrics for both runs are similar (NetworkProcessorAvgIdlePercent, RequestHandlerIdlePercent, TotalProduceTime, etc.)
- GC logs are similar in terms of object allocations and the number of collections per second and the pause times.

Conclusion

From these tests, we can conclude the following:

- We should optimize the producer for max.inflight=2. The data suggests that there is really no benefit to any other value, especially when there is low latency between the client and the broker.
- We don't understand the behavior of acks=all and acks=1 across different workloads and across the entire latency spectrum. We should leave the default as is.