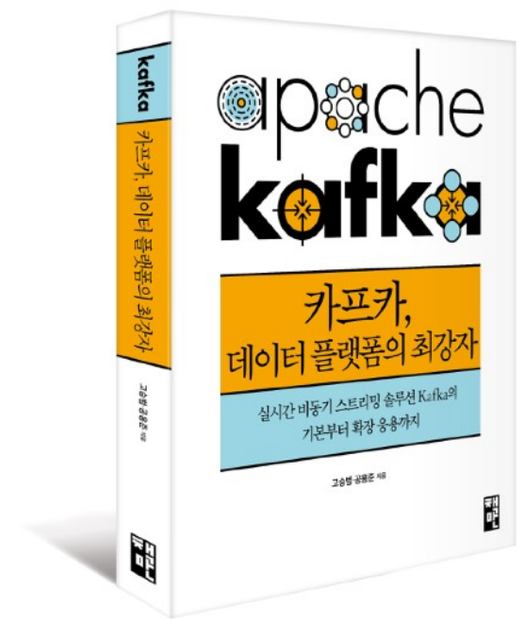
예제로 살펴보는 모니터링

2020.08.20

고승범

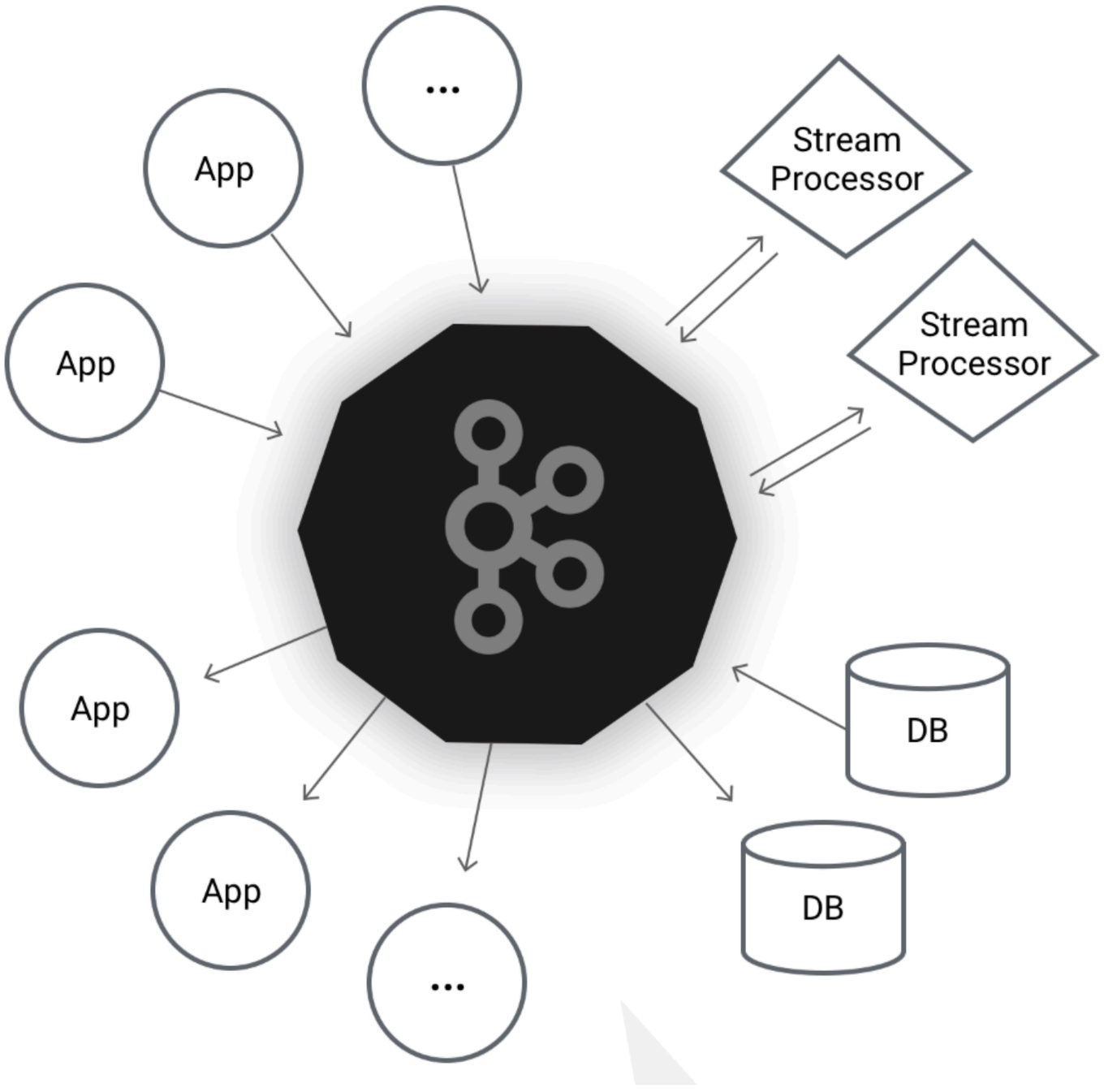
- SK텔레콤 Data Platform팀
- 페이스북 KAFKA 한국 사용자 그룹 리더

  (http://facebook.kafkakru.com)
- Confluent Certified Trainer
- Confluent Certified Administrator for Apache Kafka
- <카프카, 데이터 플랫폼의 최강자> 저자
- 예스24 IT 분야 1위 (이틀..;)
- popit.kr 저자, 개인 브런치 운영 중



목차

1. 모니터링
2. 지연 이슈
3. 지연은 왜 발생하는가?
4. 잘 보내고 있는가?
5. 잘 받고 있는가?



1. 모니터링

모니터링

- 목적
  - 카프카는 실시간 분산 이벤트 스트리밍 플랫폼
  - 중앙 허브 역할
  - 높은 성능의 파이프 라인들, 스트리밍 분석, 데이터 통합
- 무엇을?
  - 모든 메시지들을 잘 받고 있는지?
  - 유실은 없는지?
  - 지연은 없는지?
- 어떻게?
  - JMX 메트릭

모니터링툴

- JMX 모니터링
  - 카프카 전반적인 상태
  - Producer, Consumer의 전반적인 상태
  - Prometheus + JMX exporter + Grafana
  - Jconsole
- Lag 모니터링
  - Consumer의 지연
  - Burrow
  - Prometheus + Kafka exporter + Grafana
  - Kafka exporter
  - Grafana ID

cluster kafka topic All consumergroup All

Messages In Per Topic

토픽과 컨슈머 그룹 선택 가능

Bytes In Per Topic

Bytes Out Per Topic

Topic Partition Count

토픽별 파티션 수 확인

Topic Current Offset

오프셋을 이용한 토픽의 메시지 유입 여부 확인

Consumer Group Lag Count

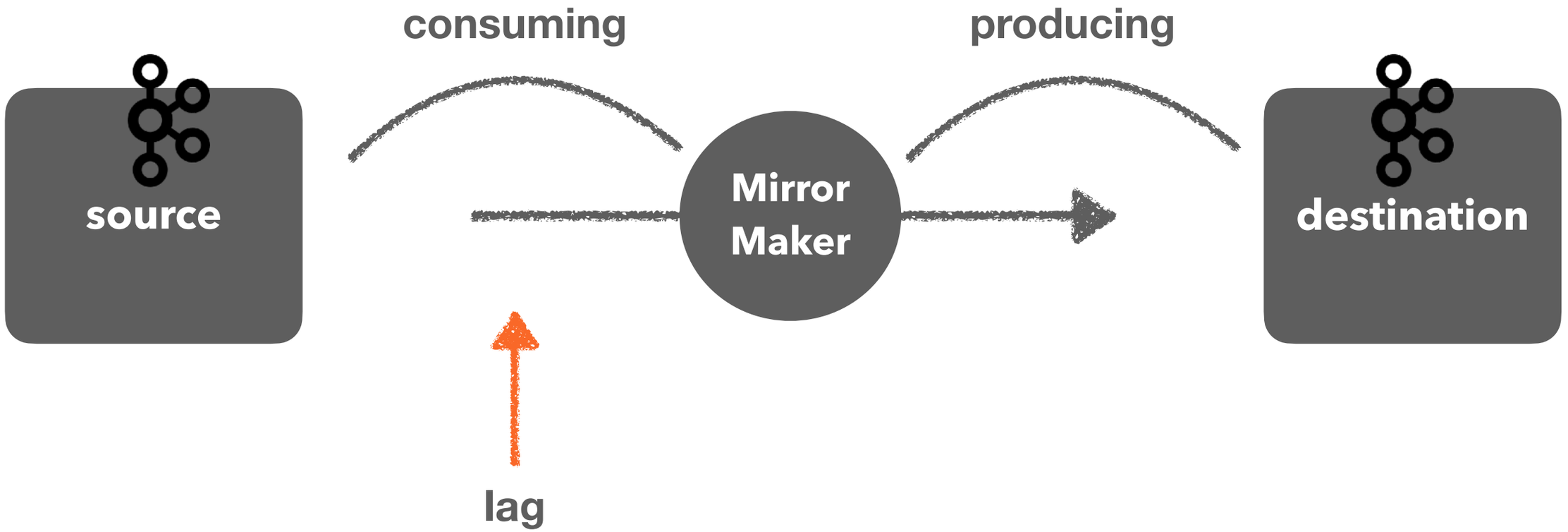
컨슈머 그룹별 LAG 카운트 확인

Consumer Group Current Offset

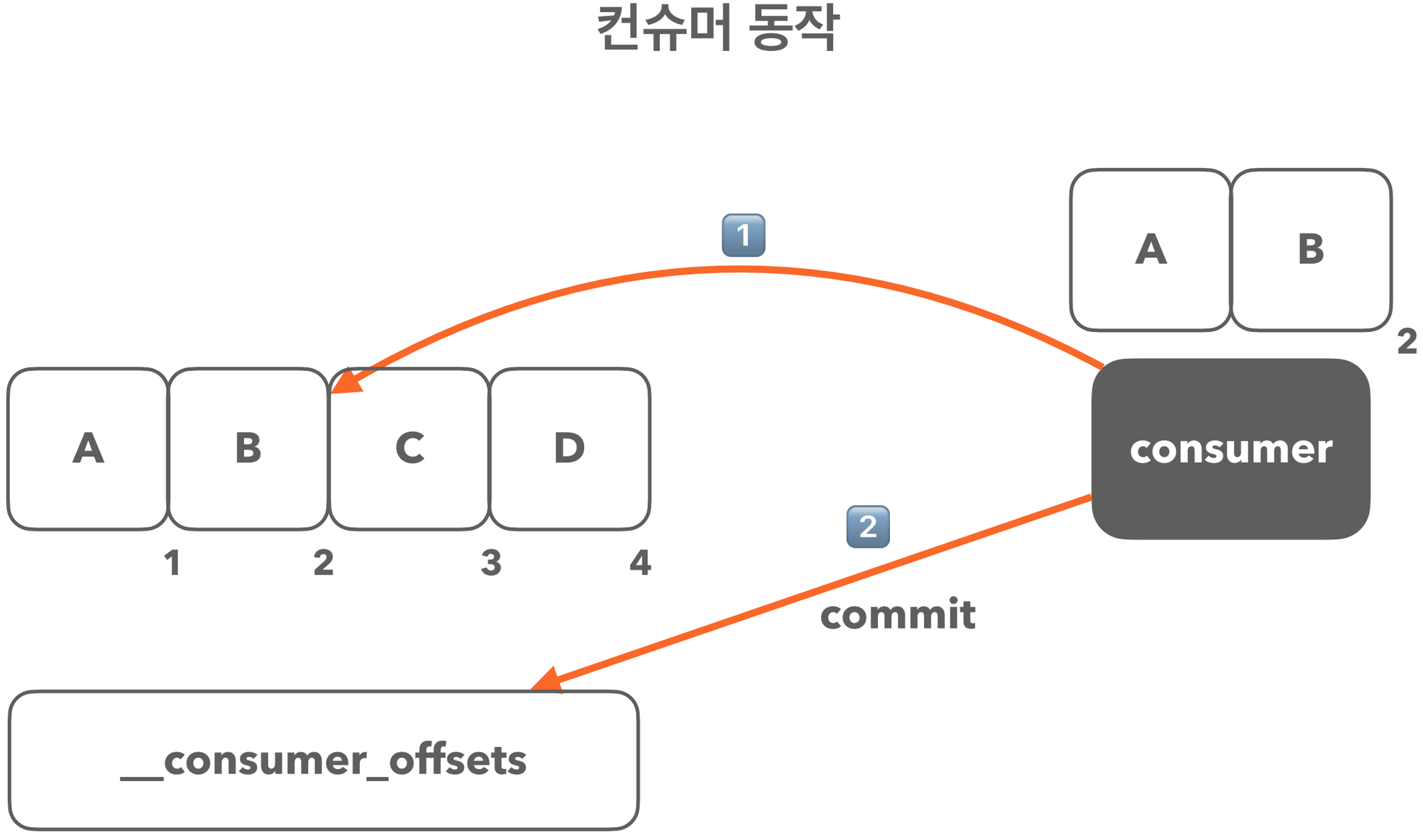
컨슈머 그룹별 토픽의 파티션 별 오프셋 확인

2. 지연 이슈

데이터 흐름



LAG이란?



Latest Offset - Last committed Offset = LAG

4 - 2 = 2

3. 지연은 왜 발생하는가?

Lag 분석

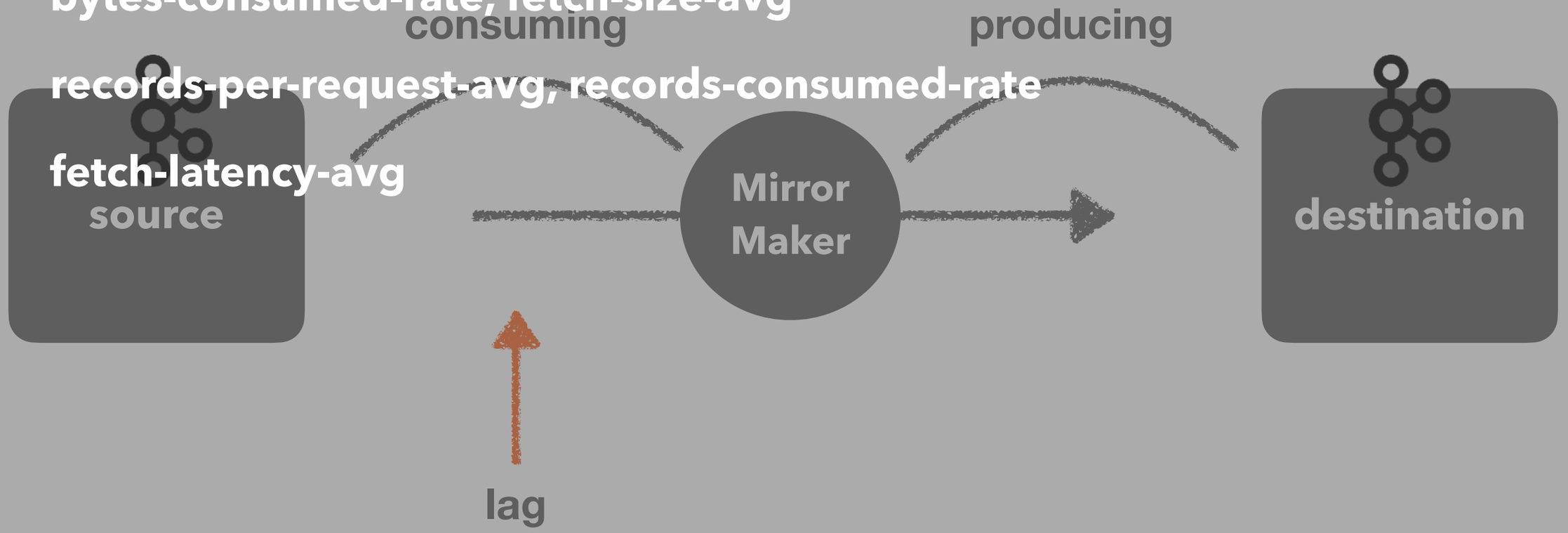
lag이 발생하는 이유는 **producing rate > consuming rate**

consumer 수를 늘려준다. -> 일반적인 해결 방법

bytes-consumed-rate, fetch-size-avg

records-per-request-avg, records-consumed-rate

fetch-latency-avg



Lag 분석

lag이 발생하는 이유는 producing rate > consuming rate

consumer 수를 늘려준다. -> 일반적인 해결 방법

bytes-consumed-rate, fetch-size-avg

Lag 분석

lag이 발생하는 이유는 producing rate > consuming rate

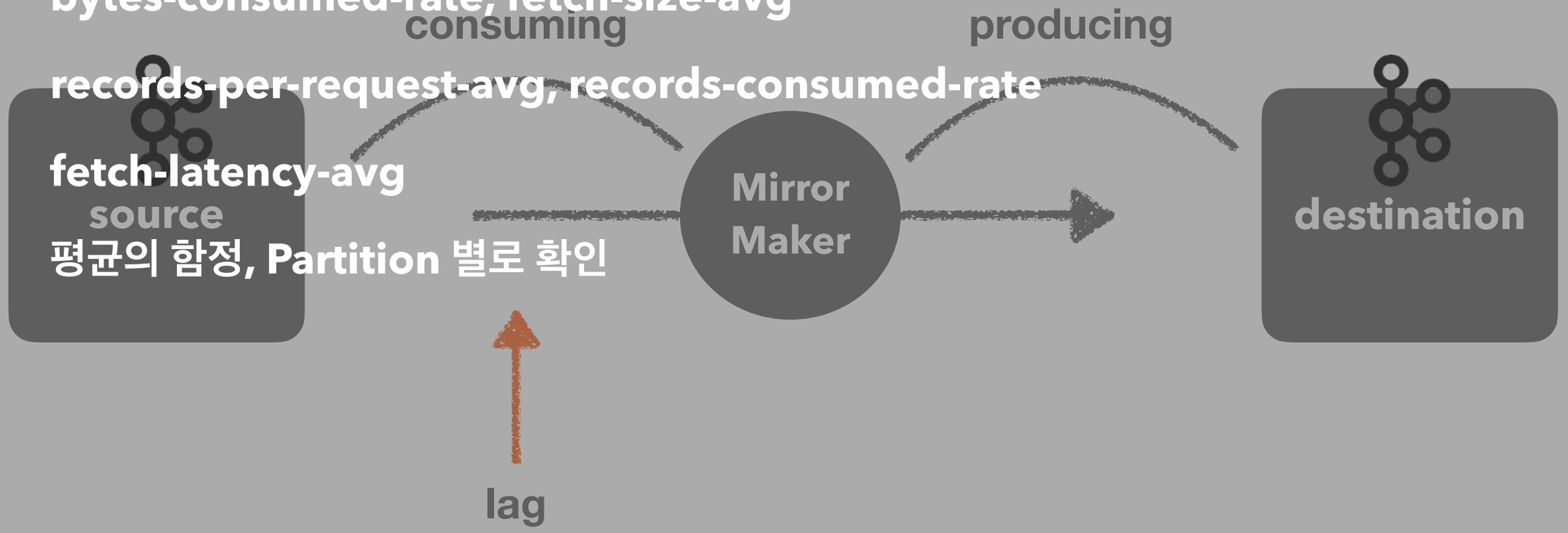
consumer 수를 늘려준다. -> 일반적인 해결 방법

bytes-consumed-rate, fetch-size-avg

records-per-request-avg, records-consumed-rate

fetch-latency-avg

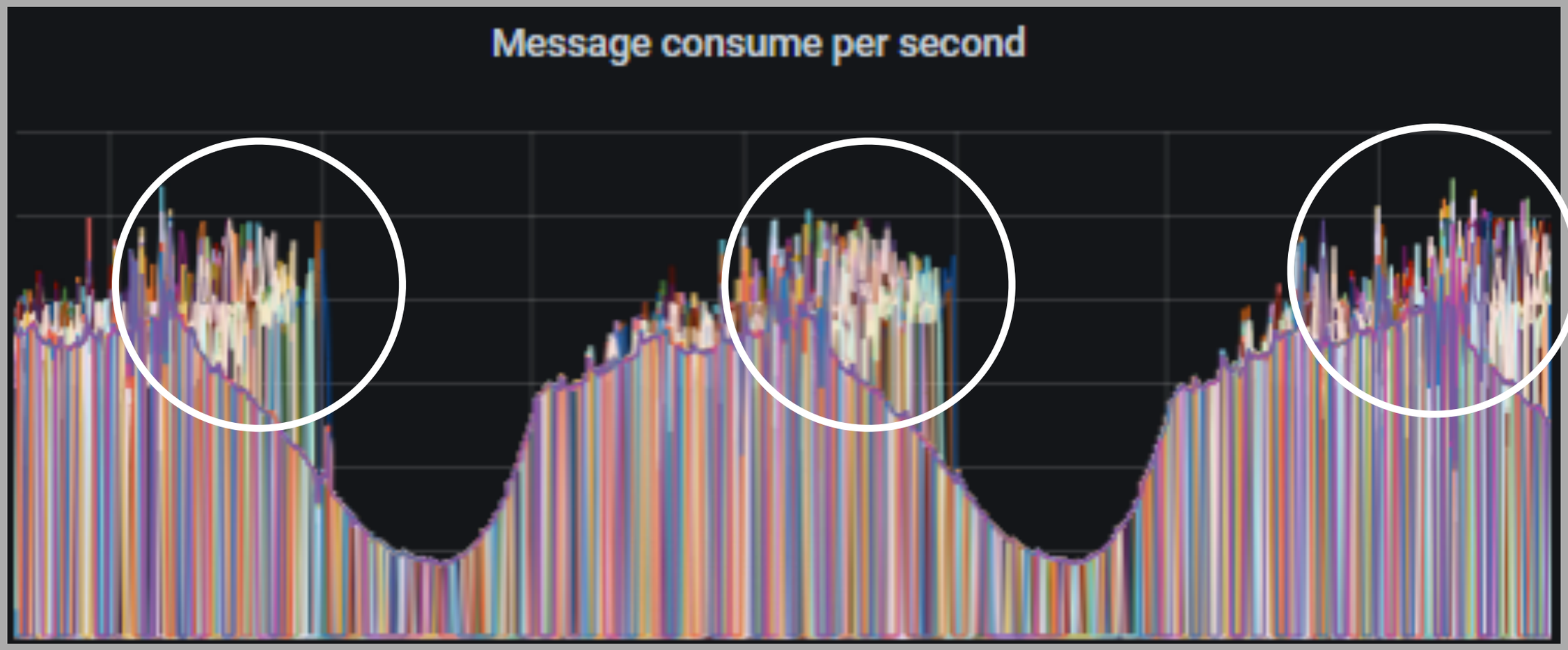
평균의 함정, Partition 별로 확인



Lag 분석

lag이 발생하는 이유는 producing rate > consuming rate

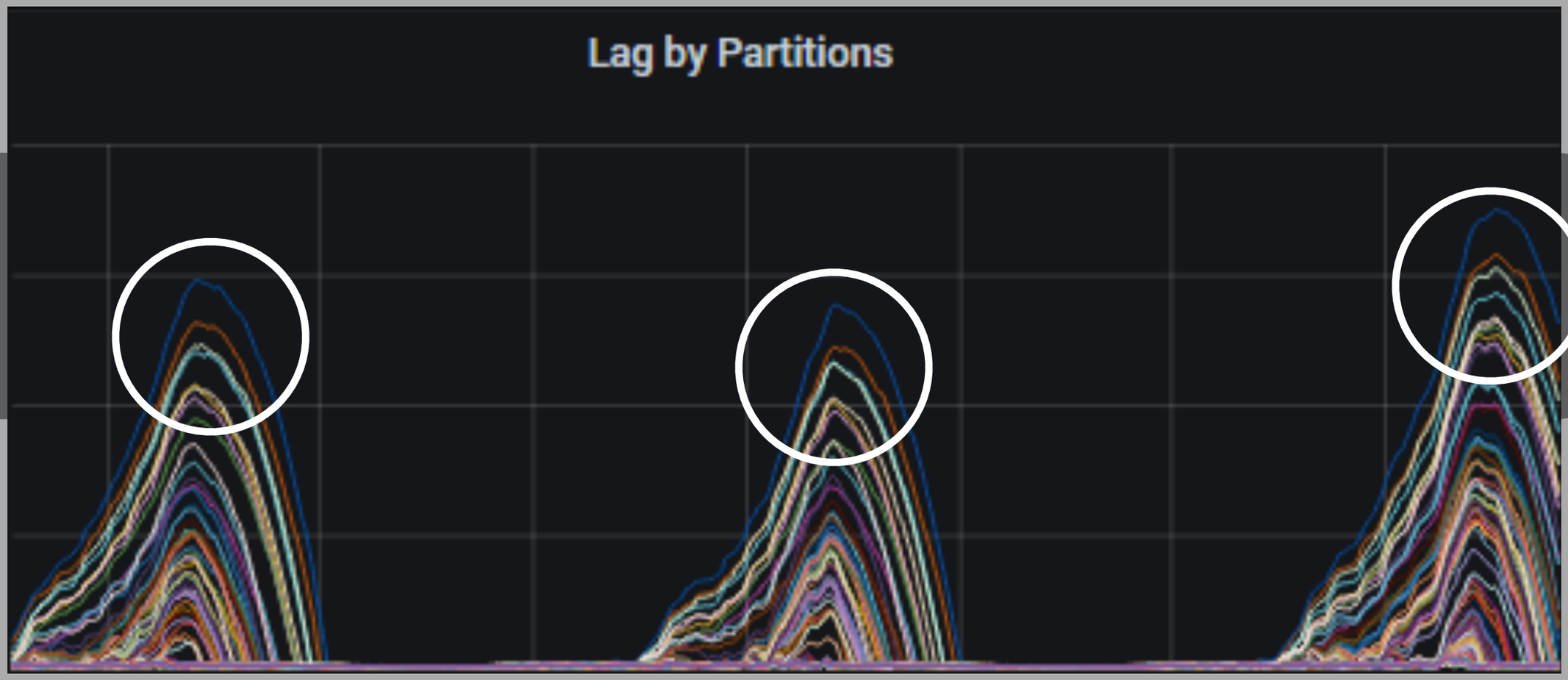
consumer 수를 늘려준다. -> 일반적인 해결 방법



Lag 분석

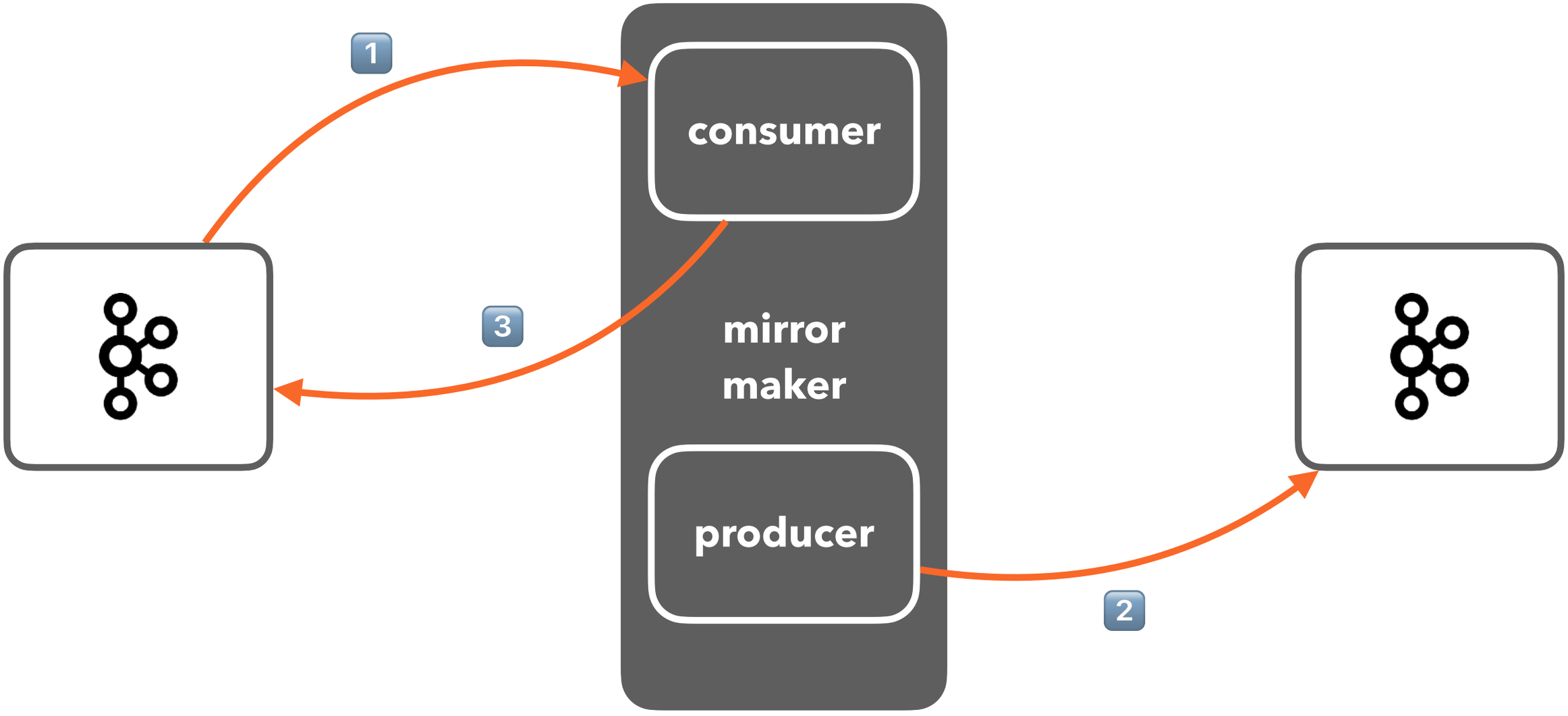
lag이 발생하는 이유는 producing rate > consuming rate

consumer 수를 늘려준다. -> 일반적인 해결 방법



4. 잘 보내고 있는가?

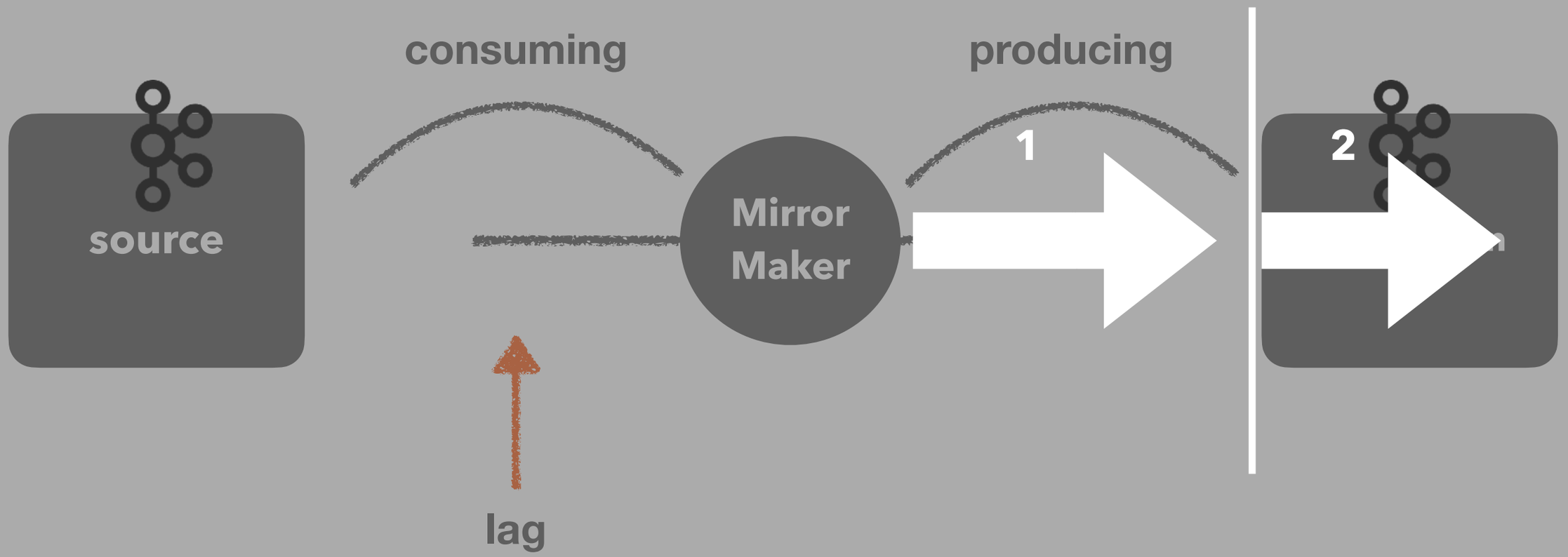
Mirror maker란?



Producer 분석

mirror maker는 잘 보내주는가? - 1

kafka는 잘 받고 있는가? - 2



Producer 분석

mirror maker는 잘 보내주는가? - 1

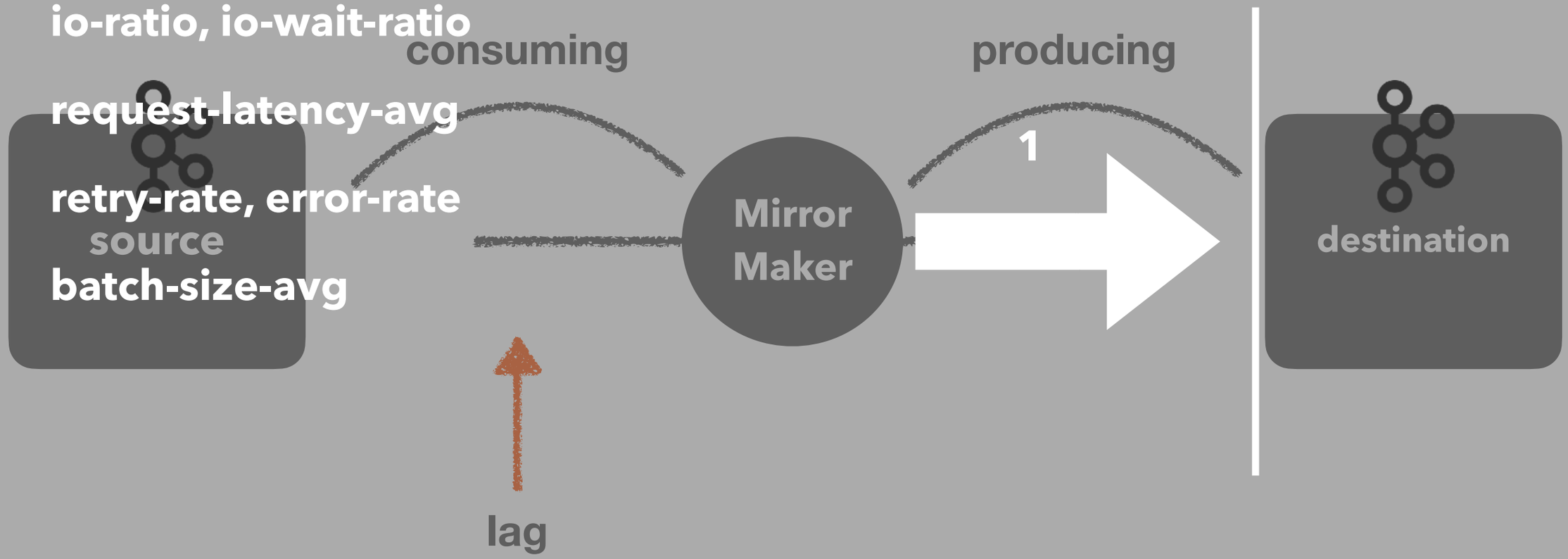
compression-rate-avg

io-ratio, io-wait-ratio

request-latency-avg

retry-rate, error-rate

batch-size-avg



Producer 분석

mirror maker는 잘 보내주는가? - 1

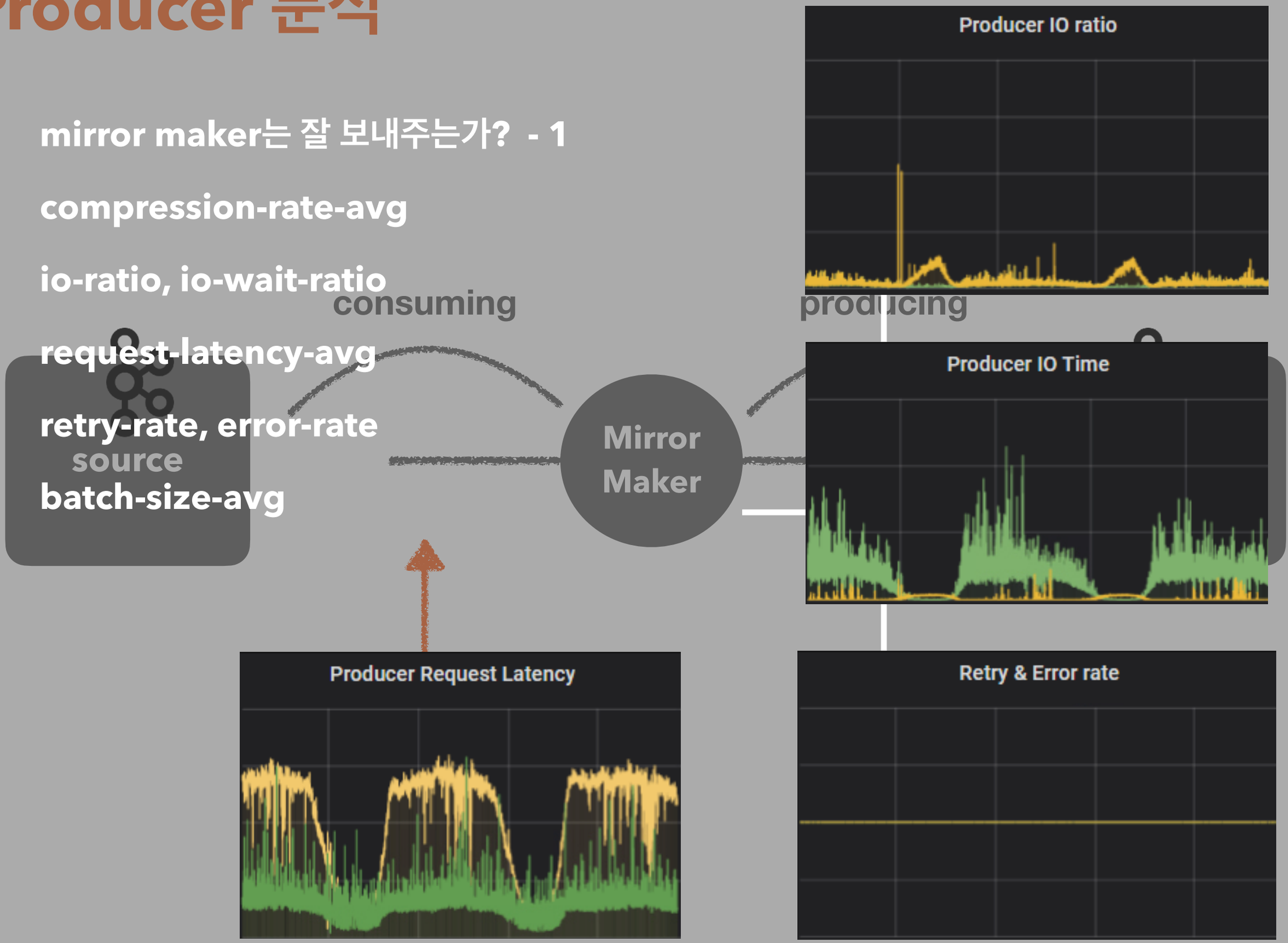
compression-rate-avg

io-ratio, io-wait-ratio

request-latency-avg

retry-rate, error-rate

batch-size-avg

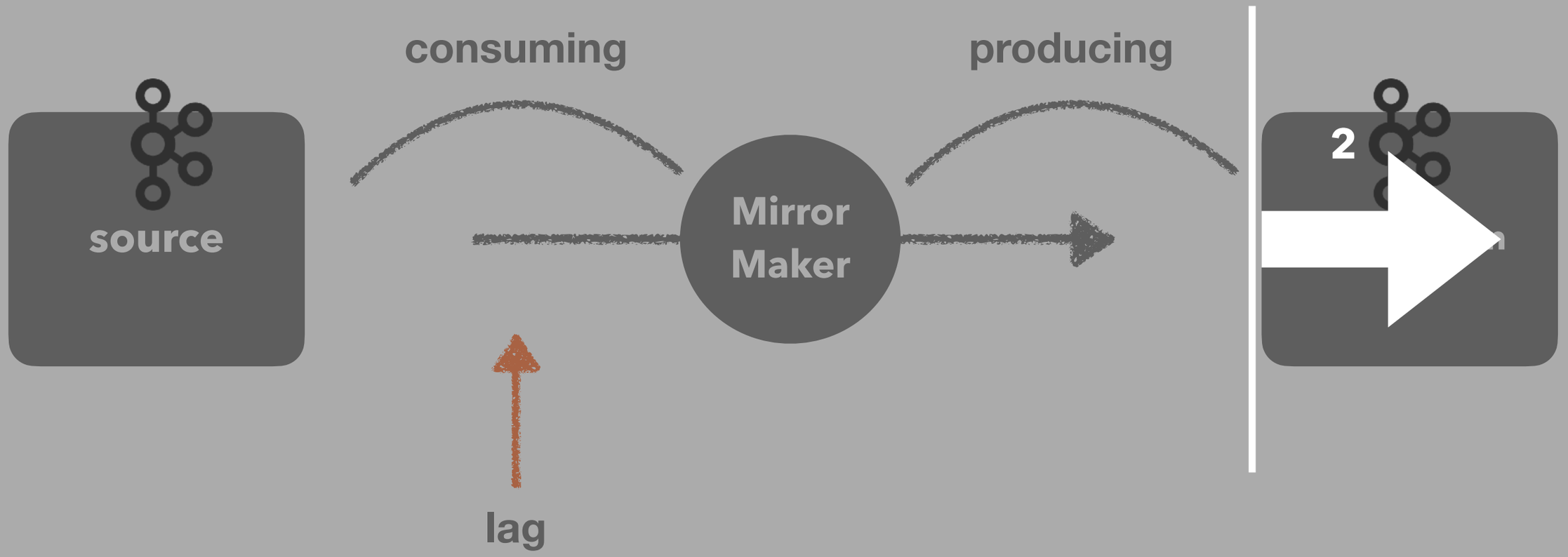


5. 잘 받고 있는가?

Kafka 분석

kafka는 잘 받고 있는가? - 2

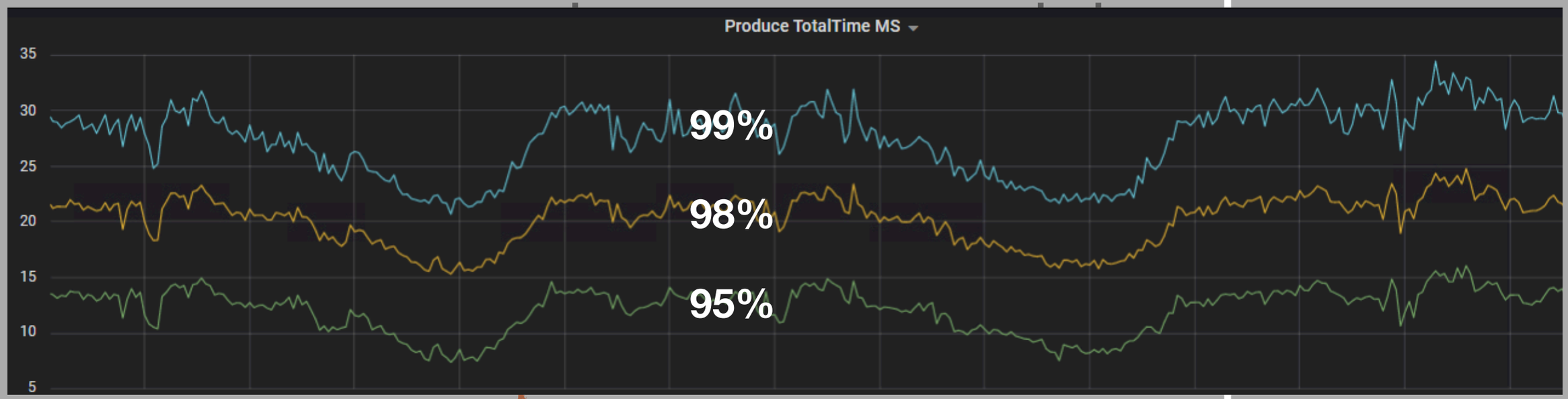
produce TotalTimeMs



Kafka 분석

kafka는 잘 받고 있는가? - 2

produce TotalTimeMs

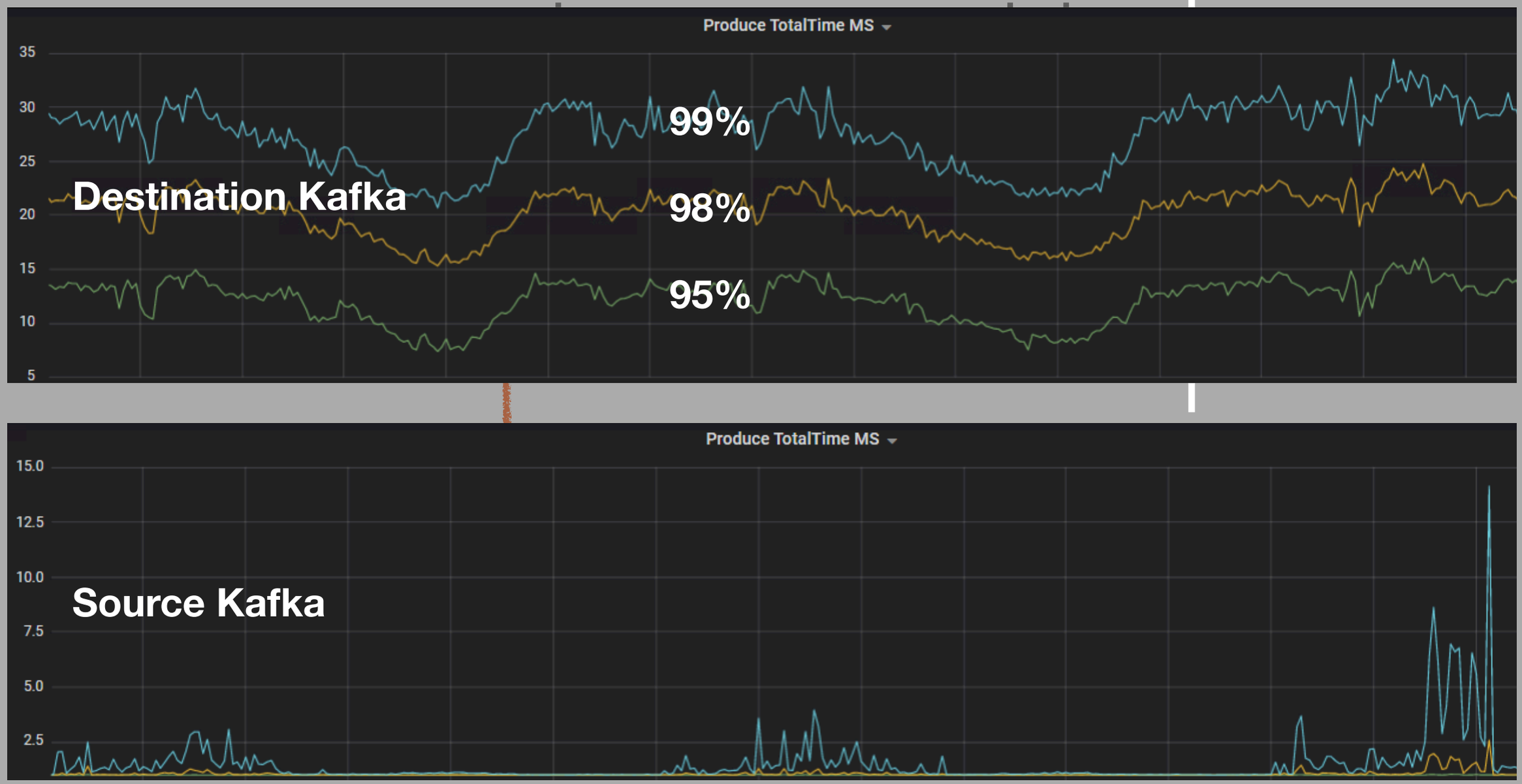


lag

Kafka 분석

kafka는 잘 받고 있는가? - 2

produce TotalTimeMs



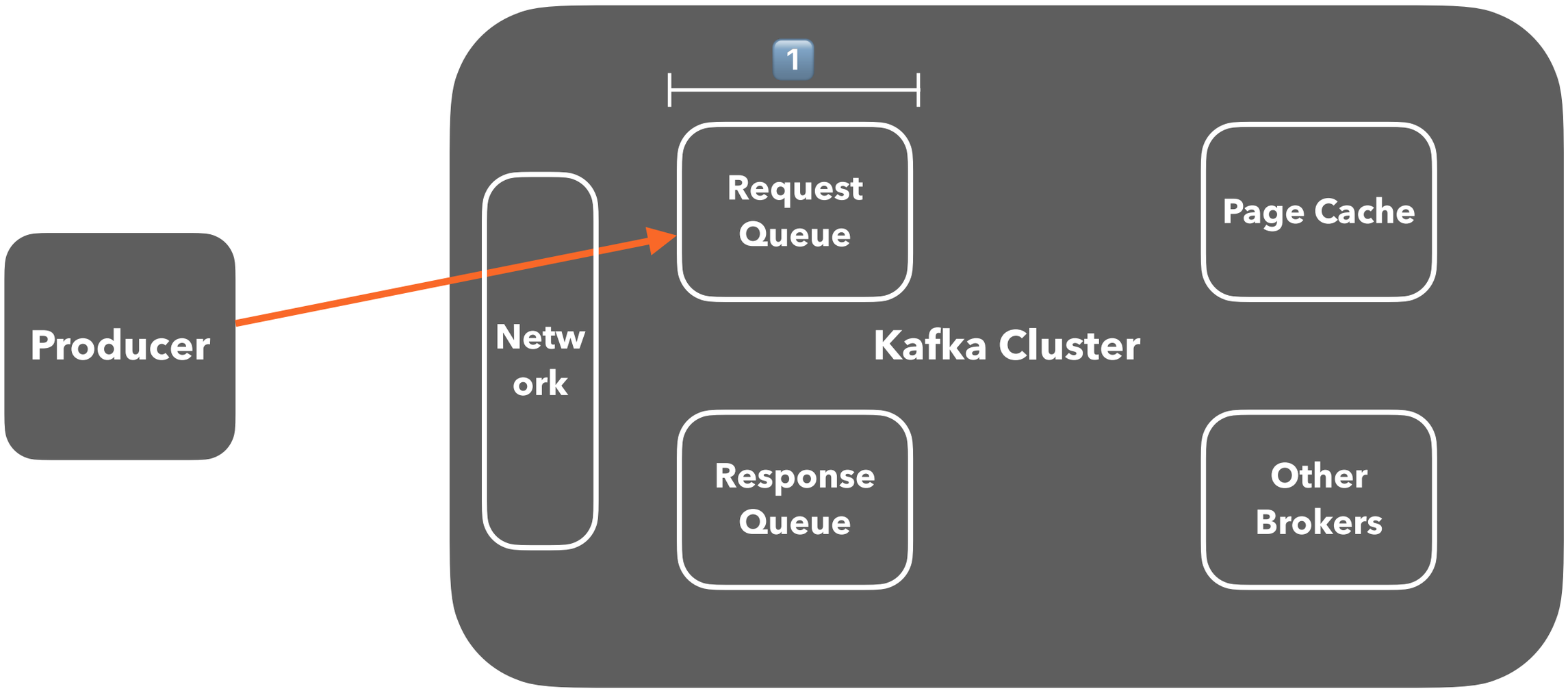
Producer Total Time?

- Request Queue Time
- Local Time
- Remote Time
- Response Queue Time
- Response Send Time

Producer Total Time?

Request Queue Time

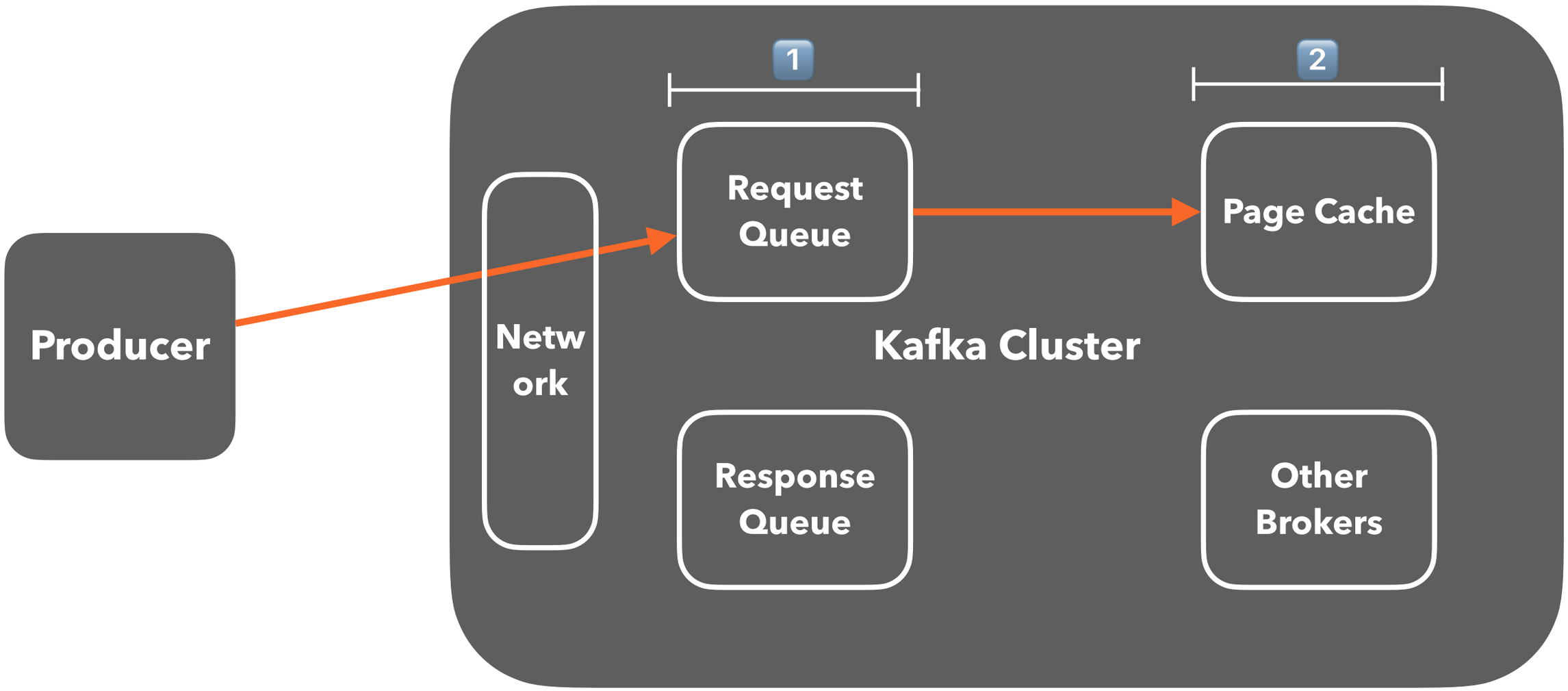
Time the request waits in the request queue



Producer Total Time?

Local Time

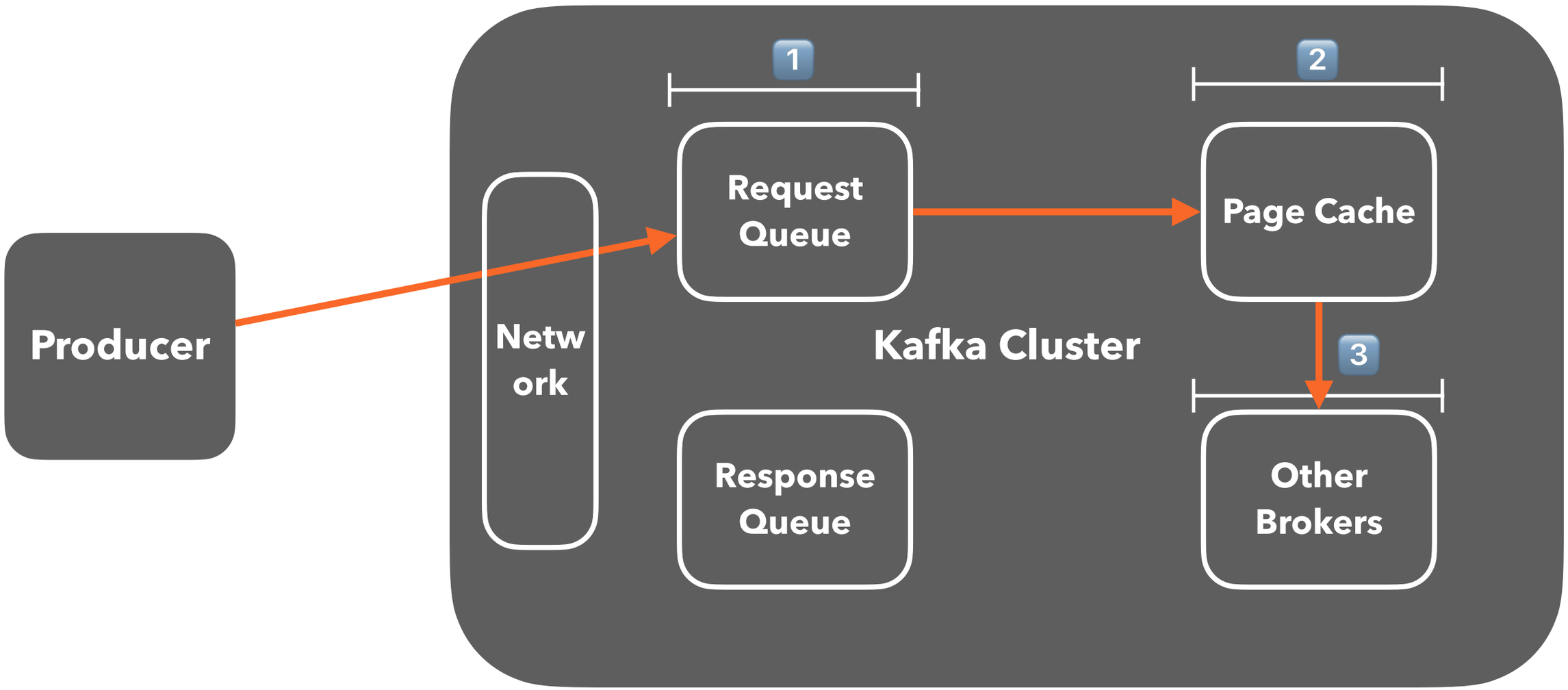
Time the request is processed at the leader



Producer Total Time?

Remote Time

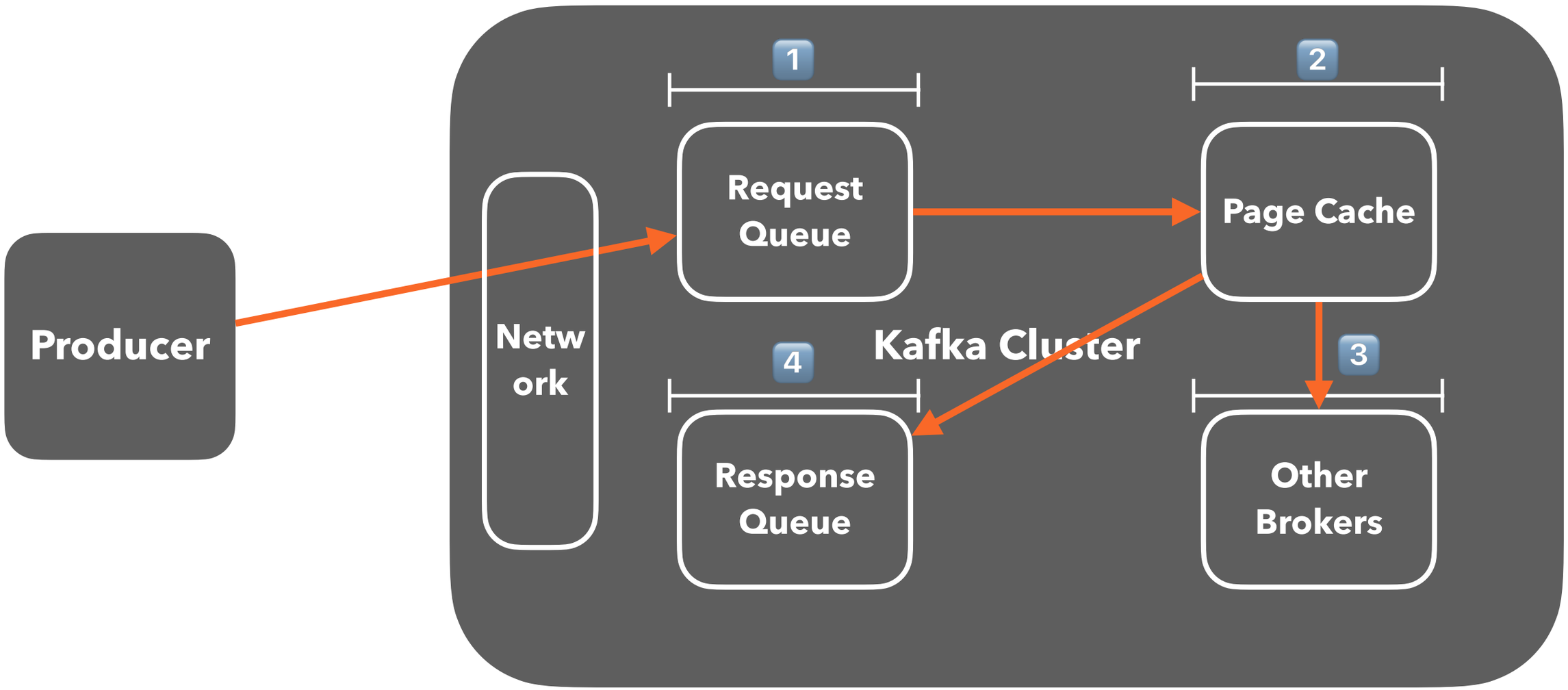
Time the request waits for the follower



Producer Total Time?

Response Queue Time

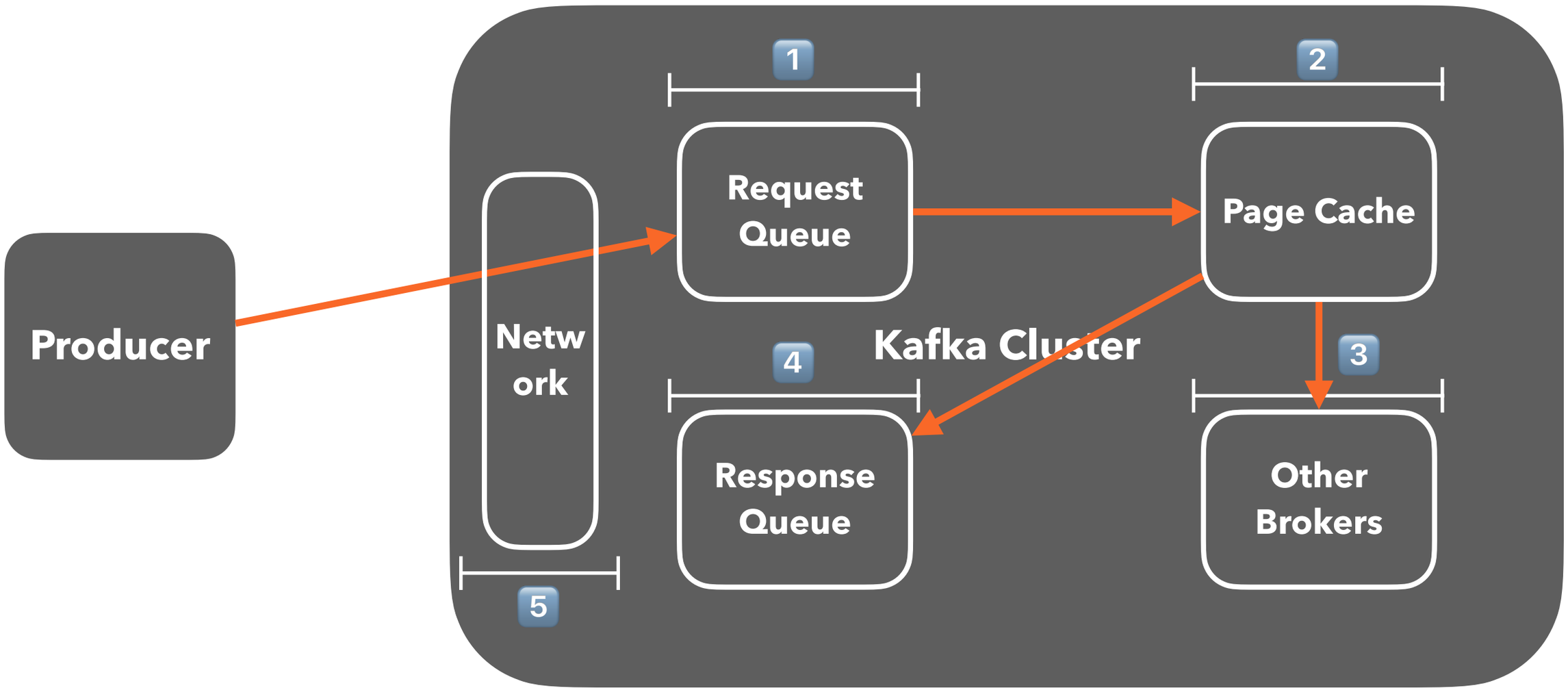
Time the request waits in the response queue



Producer Total Time?

Response Send Time

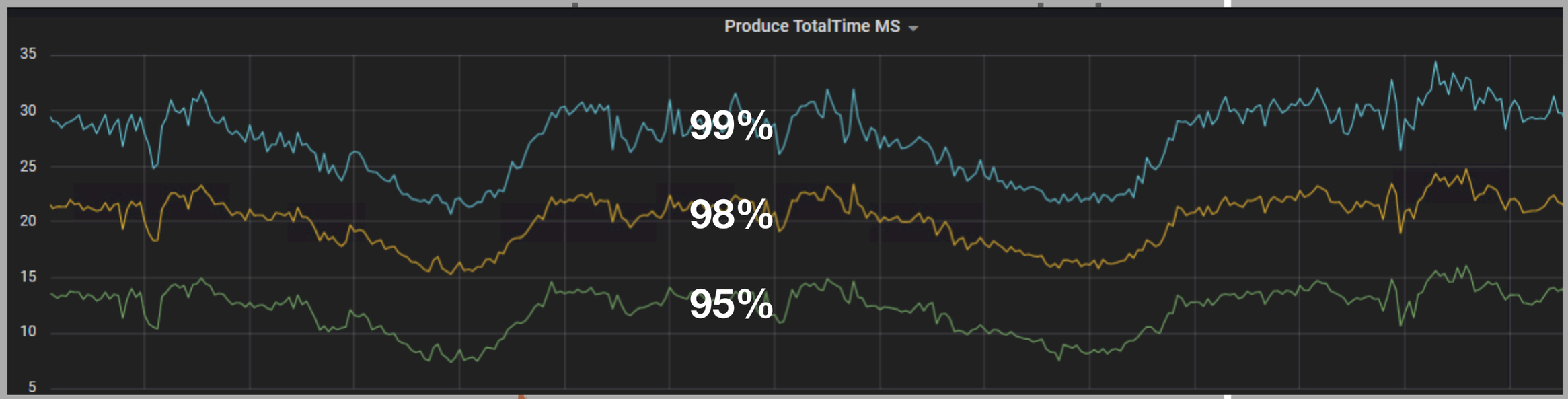
Time to send the response



Kafka 분석

kafka는 잘 받고 있는가? - 2

produce TotalTimeMs



lag

Kafka 분석

kafka는 잘 받고 있는가? - 2

produce TotalTimeMs

Kafka 분석

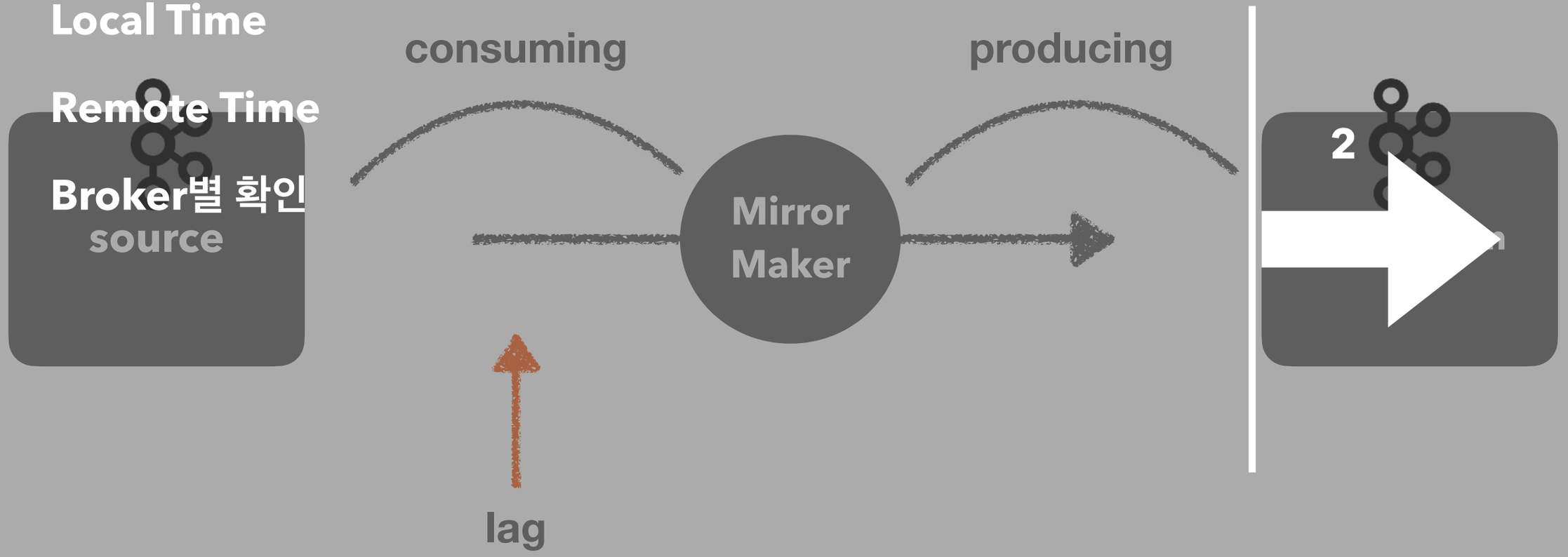
kafka는 잘 받고 있는가? - 2

produce TotalTimeMs

Local Time

Remote Time

Broker별 확인

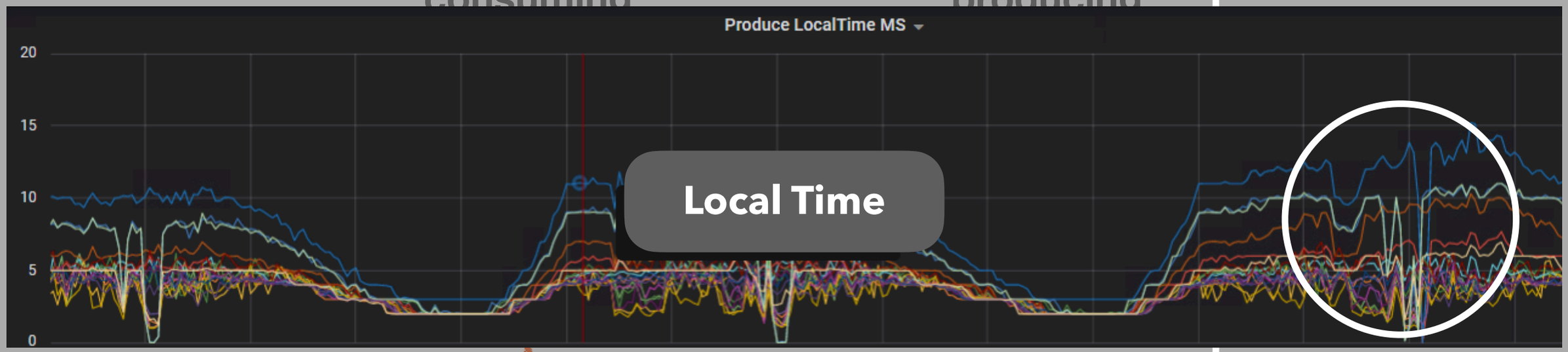


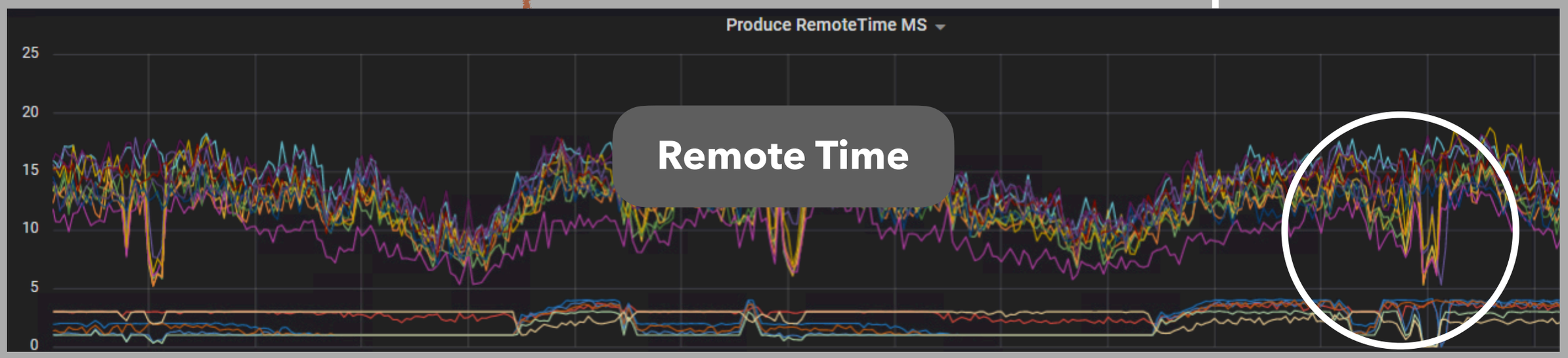
Kafka 분석

kafka는 잘 받고 있는가? - 2

produce TotalTimeMs

Local Time





결론

- 모니터링 적용
- 이슈 발생 시 전,후 구간 별 분석
- **Metric은 Broker, Partition별 확인**
- 하나의 **Consumer group**은 하나의 **Topic**만 적용

감사합니다