

[Abrir en la aplicación](#)

Jhansi Karee

[Seguir](#)

208 Seguidores

Acerca de

How to parallelise Kafka consumers

**Jhansi Karee** Jul 20, 2019 · 5 min read

Kafka is an asynchronous messaging queue. Kafka consumer, consumes message from Kafka and does some processing like updating the database or making a network call. If you are fairly new to Kafka concepts, please read my [blog on basic concepts of Kafka](#).

As we see, Kafka consumers might do some time taking operations. This means consumers might not catch up with the speed at which the messages are being produced and thus increasing lag. Lag is the number of new messages which are yet to be read.

One of the good things we get using Asynchronous messaging queues like Kafka, is producers and consumers can write and read at their own speed. But, slow processing consumers could lead to high lag in Kafka. Kafka's way of solving this problem is by using consumer groups.

What is a Consumer group?

Consumer group is a grouping mechanism of multiple consumers under one group. Data is equally divided among all the consumers of a group, with no two consumers of a group receiving the same data. Let's see more details about it.

While consuming from Kafka, consumers could register with a specific group-id to Kafka. Consumers registered with the same group-id would be part of one group.

[Abrir en la aplicación](#)

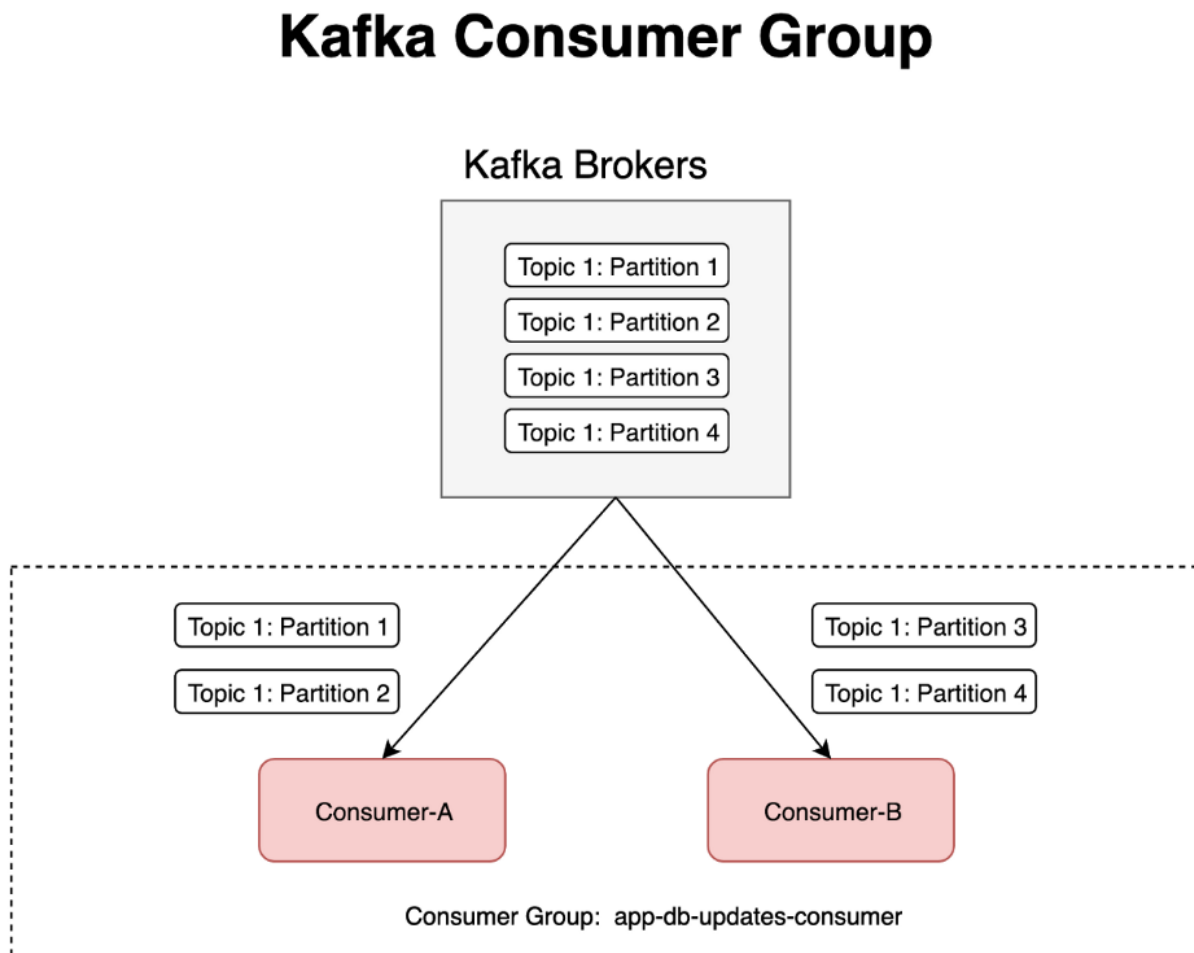
How Kafka assigns the partitions to consumers?

Before assigning partitions to a consumer, Kafka would first check if there are any existing consumers with the given group-id.

When there are no existing consumers with the given group-id, it would assign all the partitions of that topic to this new consumer.

When there are two consumers already with the given group-id and a third consumer wants to consume with the same group-id. It would assign the partitions equally among all the three consumers. No two consumers of the same group-id would be assigned to the same partition.

Suppose, there is a topic with 4 partitions and two consumers, consumer-A and consumer-B wants to consume from it with group-id “app-db-updates-consumer”.



[Abrir en la aplicación](#)

As shown in the diagram, Kafka would assign:

- partition-1 and partition-2 to consumer-A
- partition-3 and partition-4 to consumer-B.

This means, **the same data wouldn't be consumed by the consumers within the same group.**

How to decide on whether to use same or different consumer group for the consumers? It depends on use case to use case. Let's understand this in more detail.

When to use the same consumer group?

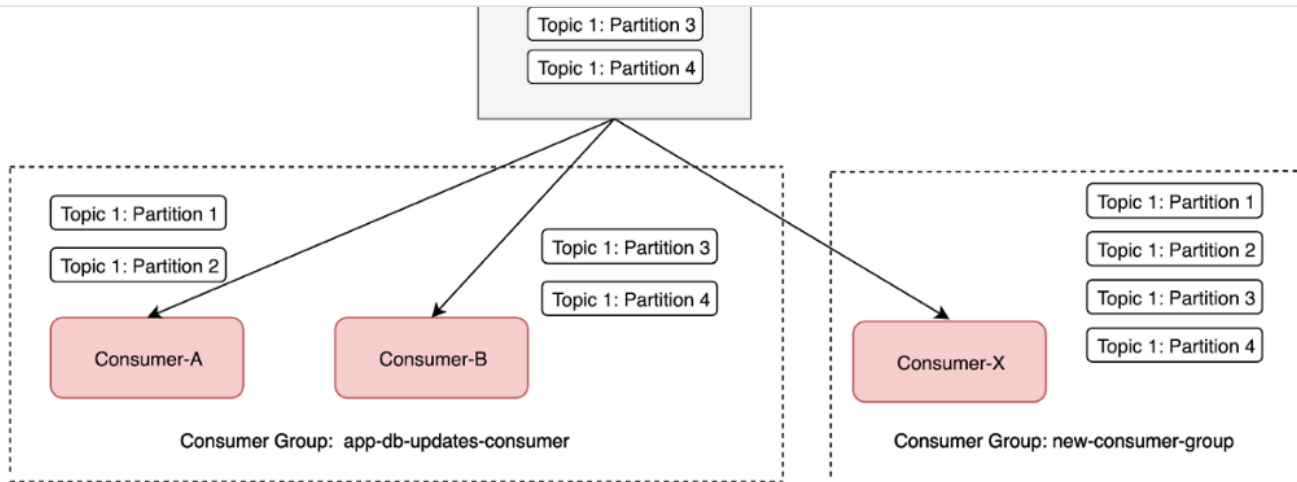
Consumers should be part of the same group, when the consumer performing an operation needs to be scaled up to process in parallel. Consumers part of the same group would be assigned with different partitions. As said before, no two consumers of the same group-id would get assigned to the same partition. Hence, each consumer part of a group would be processing different data than the other consumers within the same group. Leading to parallel processing. This is one of the ways suggested by Kafka to achieve **parallel processing in consumers.**

When to use the different consumer group?

Consumers should not be within the same group, when the consumers are performing different operations. Some consumers might update the database, while other set of consumers might do some computations with the consumed data. In this case definitely we would want all these different consumers to be reading all the data from all the partitions. Hence, in this kind of use case to read data from all the partitions, we should register these consumers with different group-id.

Multiple Kafka Consumer Groups

Kafka Brokers

[Abrir en la aplicación](#)


How would the offsets be maintained for consumers of different groups?

Offset, an indicator of how many messages has been read by a consumer, would be maintained per consumer group-id and partition. When there are two different consumer groups, 2 different offsets would be maintained per partition. Consumers of different consumer groups can resume/pause independent of the other consumer groups. Hence, leaving no dependency between the consumers of different groups.

Let me try to think some of the questions you still might have.

Let's take the same use case again. When there is a topic with 4 partitions and two consumers, consumer-A and consumer-B are already consuming from it with group-id "app-db-updates-consumer".

Q. What if consumer-B goes down?

A. *Kafka will do rebalancing and it would assign all the four partitions to consumer-A.*

P. ¿Qué pasa si los nuevos consumidores, consumidor-C y consumidor-D comienzan a consumir con el mismo ID de grupo "app-db-updates-consumer"? R. *Kafka volverá a realizar el reequilibrio y asignaría a cada consumidor una partición por igual.*

P. ¿Qué pasa si un nuevo consumidor, consumidor-E se une con el mismo ID de grupo "app-db-updates-consumer"? Esto totaliza 5 consumidores, donde las particiones son 4?

[Abrir en la aplicación](#)

P. ¿Kafka puede asignar la misma partición a dos consumidores? R. *Kafka no puede asignar la misma partición a dos consumidores dentro del mismo grupo. ¿Qué pasa entonces con los diferentes grupos de consumidores? Las particiones solo se dividen entre los consumidores del mismo grupo. Esto significa que Kafka asignará las mismas particiones a dos consumidores de diferentes grupos.*

P. ¿Cuál es el número óptimo de consumidores dentro del mismo grupo? R. *El número de consumidores dentro de un grupo puede ser como máximo el mismo número de particiones. Kafka puede, como máximo, asignar una partición a un consumidor. Si hay más consumidores que las particiones, Kafka no alcanzaría las particiones para asignar a los consumidores. No todos los consumidores del grupo serían asignados a una partición y, por lo tanto, algunos de los consumidores del grupo estarían inactivos.*

Conclusión:

Hemos visto cómo funcionan los grupos de consumidores de Kafka y cómo podríamos equiparar a los consumidores compartiendo el mismo ID de grupo. Sin embargo, con este enfoque, el escalado de los consumidores no puede ir más allá del número de particiones. ¿Podemos paralelizar a los consumidores de Kafka más allá del número de particiones? [Lea mi blog sobre cómo lograr esto](#) .

[Big Data](#)[Kafka](#)[Consumidor](#)[Procesamiento en paralelo](#)[Grupo de consumidores Kafka](#)

Acerca AyudaLegal
de

Obtén la aplicación Medium

[Abrir en la aplicación](#)

