Kafka Fundamentals Brain Teasers: M4: How Kafka Works

Overview

Key concepts from Module 4 are:

- A simple producer code example in the Java API
- A simple consumer code example in the Java API
- Replication, leaders, and followers
- Deletion retention policy
- Producer design
- Acknowledgments
- Idempotence
- Partitioning strategies
- Consumer groups and rebalancing
- Security

Here's the quick quiz on Module 4 (https://forms.gle/JyY2w9FN6iCTsp5y7) from the Online Talk Series.

Problem #4A: Relating Consumers in Groups

Suppose you have a topic with 3 partitions, p_0, p_1, and p_2. Further, suppose we have consumer group g_0 with consumers c_0 and c_1. No matter what further configuration you have, what is the same about c_0 and c_1? What is different?

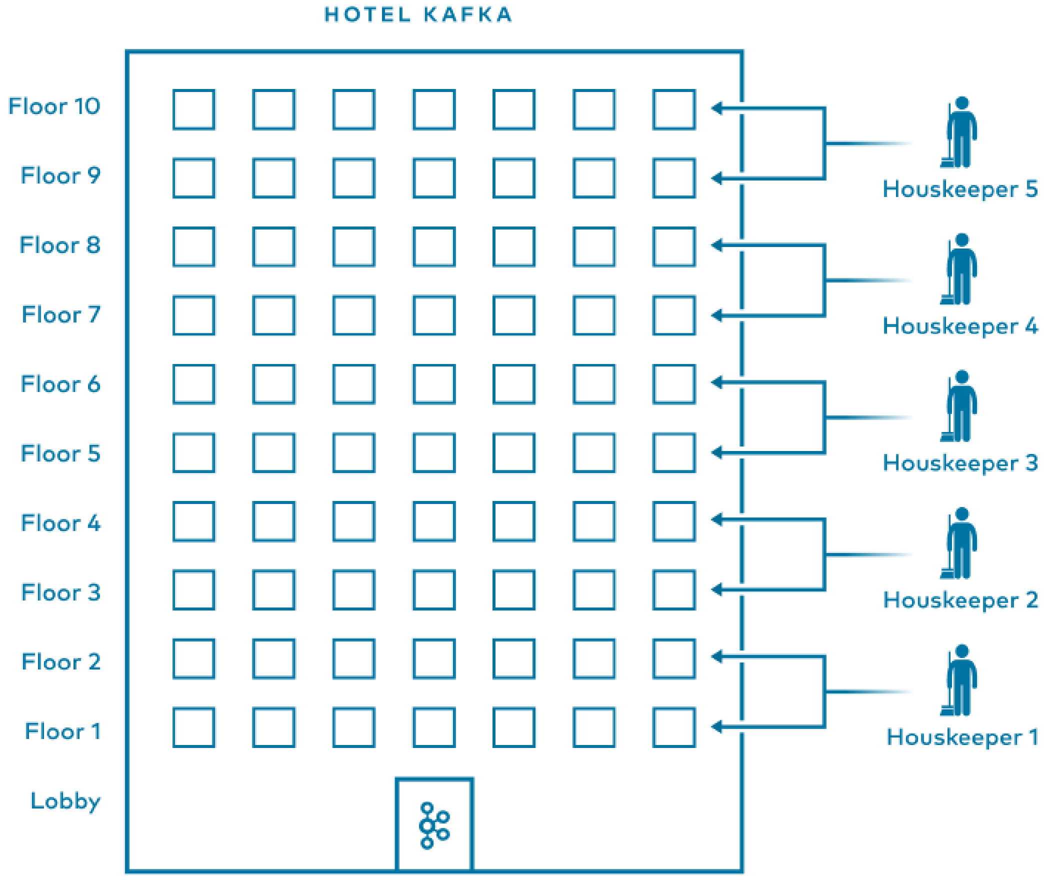
Solution #4A: Relating Consumers in Groups

In this case, both c_0 and c_1 are in the same consumer group. This means that they are performing the same logic as each other. But what is different is they are working with *different data*. The goal of a consumer group is to read all of the data from all of the topics to which its consumers are subscribed (and if all consumers in a group share the same logic, they share their topic subscription). Let's simplify it this way: These consumers are subscribed to one topic, and these consumers are working to consume all of the data from all of the partitions of that topic.

In a non-Kafka scenario, it's as if we have a hotel with 10 floors of guest rooms. We might have a team of five housekeepers, each of whom must fully clean every assigned guest room alone (no teamwork is allowed within a single room in this scenario). No matter what, by the time it's time for guests to arrive, all rooms on all 10 floors must be cleaned.

- Housekeeper 1 on the team takes Floors 1 and 2 and cleans all the dirty rooms on those floors
- Housekeeper 2 takes Floors 3 and 4 and cleans all the dirty rooms on those floors
- This continues, i.e., Housekeeper 3 cleans Floors 5 and 6, Housekeeper 4 cleans Floors 7 and 8, and Housekeeper 5 cleans Floors 9 and 10
- The housekeepers are a team, and they trust each other to perform the work

There is a division of labor wherein everyone on the team is doing the same task but on different "data." In this metaphor, the consumers are the housekeepers, and their team is the consumer group. The hotel rooms are messages, and each floor is a partition. Each housekeeper (consumer) is assigned two partitions (floors).



Problem #4B: Fixing a Broken Consumer/Partition Assignment

Suppose you have a topic with 3 partitions, p_0, p_1, and p_2. Further, suppose you have consumer group g_0 with consumers c_0 and c_1. Suppose at one point in time, we have only the following assignments: c_1 is consuming from p_0 and c_1 is consuming from p_2. What is the downside about this situation? Propose a fix.

Solution #4B: Fixing a Broken Consumer/Partition Assignment

Now, we have this assignment of consumers to partitions:

- c_1 is consuming from p_0
- c_1 is consuming from p_2

But we have three partitions: p_0, p_1, and p_2.

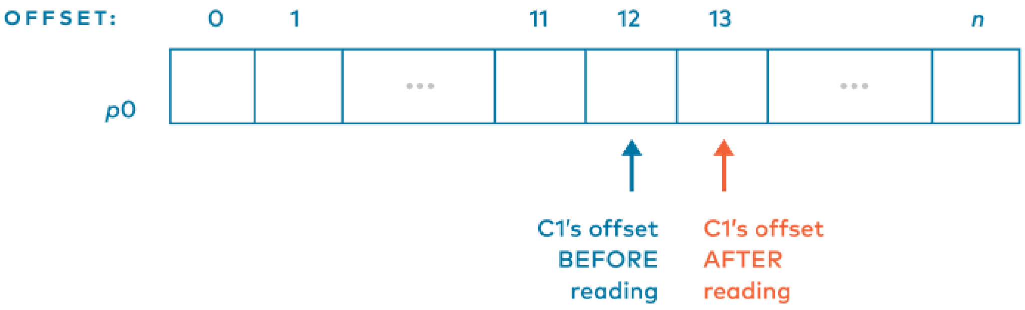
The consumer group as whole must be working together to consume from all partitions. Poor p_1 isn't getting any attention. That means all the ride requests on p_1 aren't getting matched to drivers and people are waiting in the rain. Or a bunch of food orders on p_1 aren't getting prepared and customers with limited lunch breaks will show up to find they have to wait another 15 minutes. Or all the hotel rooms on Floor 3 aren't getting clean and people are going to find a mess upon check-in. We better consume from all the partitions. The easiest fix is to have c_0 consuming from p_1.

Problem #4C: Understanding Consumer Offsets

Suppose you have a topic with 3 partitions, p_0, p_1, and p_2. Further, suppose you have consumer group g_0 with consumers c_0 and c_1. It's the same situation as before. Suppose c_1 just read the message at offset 12 in p_0. What is its consumer offset for this partition? With your fix in mind, are there any other consumer offsets stored?

Solution #4C: Understanding Consumer Offsets

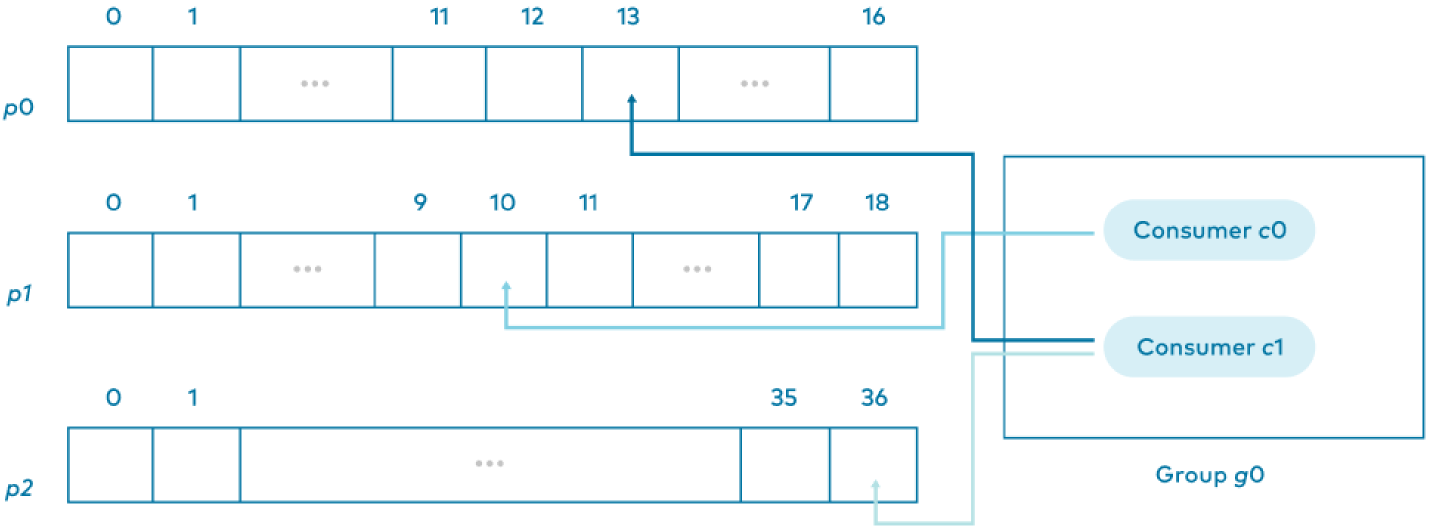
We first have the case that c_1 just read the message at offset 12 in p_0. Remember, consumers just read messages, leave them, and advance their offsets. So, c_1 will set its offset in p_0 to 13.



Think of it like a bookmark. If you just read Page 12 of a book and need to put the book down, you're most likely going to put your bookmark at Page 13 to remind you where to pick up next time. If you'd prefer to put your bookmark where you last read, that's reasonable. But the designers of Kafka had to make a decision and they made the decision that the consumer offset will tell the offset of the message to read next.

Are there other consumer offsets? Absolutely. Each consumer knows where it will read next in each partition to which it is assigned.

- c_0 has a consumer offset for where to read next in p_1
- c_1 has a consumer offset for where to read next in p_2



Back to the bookmark analogy: c_0 is reading from p_0 and p_2. Maybe you've been reading two books at once. You are likely not on the same page in both of them. A consumer needs an offset for each individual partition, just like you need a separate bookmark for each book.

Problem #4D: When Can Two Consumers Consume the Same Partition?

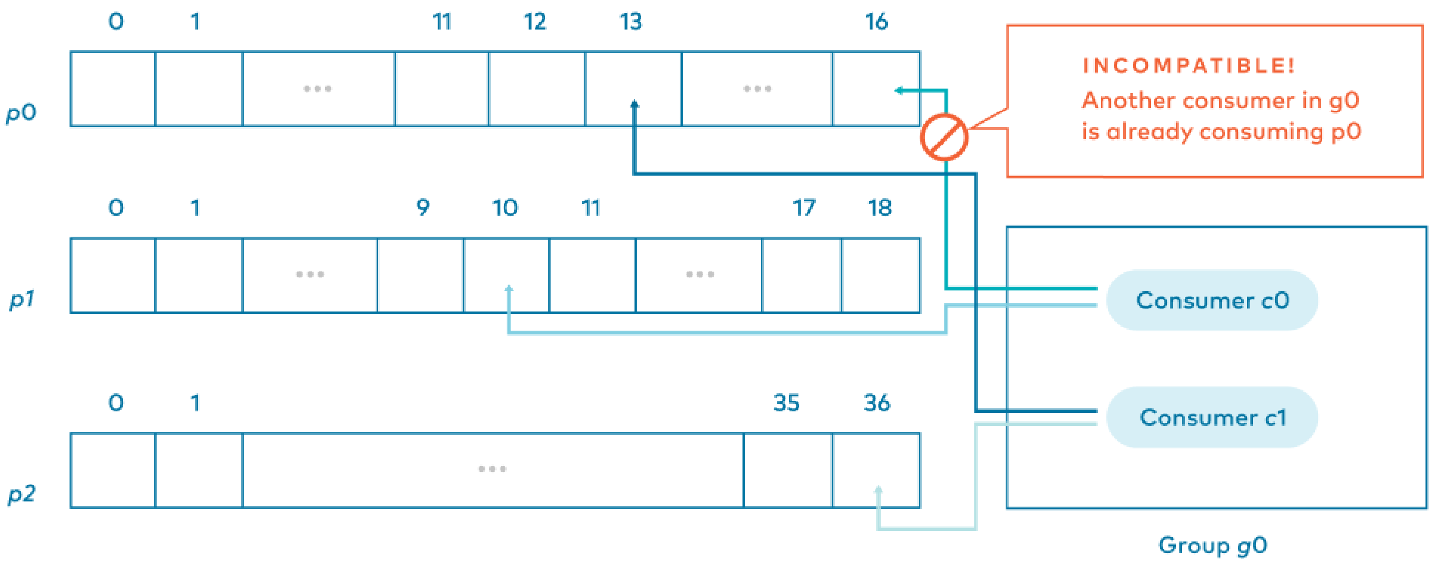
Suppose you have a topic with 3 partitions, p_0, p_1, and p_2. Further, suppose you have consumer group g_0 with consumers c_0 and c_1. You know that c_1 is consuming from p_0. Can c_0 consume from p_0? If so, why? If not, how can you change the setup to allow another consumer to consume from p_0?

Solution #4D: When Can Two Consumers Consume the Same Partition?

c_1 is consuming from p_0, and we propose c_0 is also consuming from p_0.

Before going into the answer, let's go back to that hotel rooms and housekeepers metaphor. This is like having Housekeeper 1 assigned to clean all of the rooms on Floors 1 and 2 (as stated above), but another housekeeper, such as Housekeeper 6, also assigned to clean all of the rooms on Floor 1. Housekeeper 6's shift starts after Housekeeper 1 is done for the day. What's going to happen? Housekeeper 6 goes into every room on Floor 1 and finds it clean already? Could Housekeeper 6 re-clean each room? Maybe. But cleaning products, time, effort, and laundry resources are being wasted.

In the Kafka world, the answer would be no. If two consumers in the same group were to consume from the same partition, that would mean reprocessing messages in exactly the same way. At best, that wastes resources and could bother customers with duplicate notifications or ads. At worst, a stakeholder loses money or resources. Simply put, Kafka doesn't not allow more than one consumer in a consumer group to consume from the same partition.



Note that this doesn't go both ways. A single consumer in a group could consume from more than one partition.

How can we change the problem setup to allow c_0 also to consume from p_0? c_0 can be in a different group. This way, it's working the data differently from p_0, not reprocessing it in the same way. If this sounds like a lot to manage, the good news is that Kafka takes care of all consumer/partition assignments for you. It won't let you break the rules. It'll even adjust things when a component goes down.