

Apache Kafka: Consumer Groups & Offsets

1. Consumer Groups (Scaling Reading)

Jab humein data read karna hota hai, toh ek akela Consumer kaafi nahi hota agar data bahut zyada hai. Humein "Scale" karna padta hai.

- **Concept:** Hum multiple consumers ko ek group mein daal dete hain, jise **Consumer Group** kehte hain.
- **Golden Rule:** Ek Partition ko ek time par Consumer Group ka **sirf ek hi Consumer** read kar sakta hai.
- **Goal:** Group milkar pure topic ko parallel mein read karta hai.

2. Scenarios: Consumers vs Partitions

Relation samajhne ke liye chaliye assume karte hain humare paas **Topic A** hai jisme **3 Partitions (0, 1, 2)** hain.

Case A: Balanced Load

- **3 Consumers** in Group.
- Har Consumer ko ek Partition milega.
- Ye sabse best scenario hai.

Case B: Consumers < Partitions (Less Consumers)

- **2 Consumers** in Group.
- Consumer 1: Partition 0 padhega.
- Consumer 2: Partition 1 aur 2 dono padhega.
- (Ye bhi perfectly fine hai, ek consumer zyada kaam karega).

Case C: Consumers > Partitions (Inactive Consumers)

- **4 Consumers** in Group (lekin partitions sirf 3 hain).
- Consumer 1, 2, 3 ko ek-ek partition mil jayega.
- **Consumer 4 INACTIVE (Standby) rahega.**
- **Note:** Consumer 4 partition share nahi karega. Wo bas khali baithega jab tak koi aur fail na ho jaye.

Text Diagram: The "Inactive" Scenario

```
Topic A (3 Partitions)    Consumer Group (4 Consumers)
|-- Part 0 -----> [ Consumer 1 ] (Active)
|
|-- Part 1 -----> [ Consumer 2 ] (Active)
|
|-- Part 2 -----> [ Consumer 3 ] (Active)
                        [ Consumer 4 ] (Inactive / Idle)
                        (Kyunki koi partition bacha hi nahi)
```

3. Multiple Consumer Groups

Kya hum ek hi Topic par multiple groups chala sakte hain? **Haan, bilkul.**

- Har Consumer Group independent hota hai.
- **Example (Trucks Logic):**
 - **Group 1 (Location Service):** Ye Trucks ka GPS data padh raha hai map update karne ke liye.
 - **Group 2 (Notification Service):** Ye wahi same GPS data padh raha hai SMS bhejne ke liye.
- Dono groups saare partitions ka data padhenge, apne-apne speed par.
- Group ki pehchaan `group.id` property se hoti hai.

4. Consumer Offsets (The "Bookmark" Concept)

Kafka ko kaise yaad rehta hai ki Consumer ne kahan tak data padh liya hai?

- **Problem:** Agar Consumer crash ho jaye aur wapas aaye, toh wo shuru se data nahi padhna chahta. Wo wahin se start karna chahta hai jahan usne chhoda tha.
- **Solution:** Consumers apne **Offsets Commit** karte hain.
- **Internal Topic:** Kafka ek special internal topic use karta hai jiska naam hai `__consumer_offsets` (double underscore).
 - Jab Consumer data process kar leta hai, wo Kafka ko batata hai: "Maine Offset 4262 tak padh liya."
 - Agar Consumer restart hota hai, toh Kafka `__consumer_offsets` check karta hai aur wahin se data bhejna shuru karta hai.

Text Diagram: Committing Offsets

```
[ Partition 2 ]
Msg 4260
Msg 4261
Msg 4262 <-- Consumer ne yahan tak padha aur "Commit" kiya.
Msg 4263
...

(Agar Consumer abhi fail ho jaye, toh restart hone par
Kafka usse Msg 4263 se data dega, kyunki 4262 committed hai)
```

5. Delivery Semantics (Commit kab karna hai?)

Aap Offsets kab commit karte hain, usse decide hota hai ki data loss hoga ya duplicate hoga. Iske 3 modes hain:

Mode	Behavior	Risk
At Least Once (Default)	Message process karne ke baad commit karo.	Agar process hone ke baad lekin commit hone se pehle crash hua, toh message dubara aayega (Duplicate). System ko <i>Idempotent</i> banana padega.
At Most Once	Message milte hi turant commit kar do (process karne se pehle).	Agar processing mein error aa gaya, toh wo message kabhi wapas nahi aayega (Data Loss).
Exactly Once	Message sirf aur sirf ek baar process hoga.	Sabse safe lekin complex. (Requires Kafka Streams or Transactional API).

Summary Example:

Imagine aap ek **Video Streaming App** (Netflix clone) bana rahe hain.

- Aapka **Consumer Group** users ki watching history read kar raha hai.
- Agar aap **3 Partitions** rakhte hain aur **4 Servers (Consumers)** chalte hain, toh 1 Server free baithega.
- Agar Server crash ho jata hai, toh `__consumer_offsets` ki wajah se user ki video wahin se resume hogi jahan usne chhoda tha (not from start).

Next Step: Would you like to proceed to **Kafka Brokers, Cluster Setup, and Zookeeper** detailed notes?