Kafka Interview - Day 2 (April 29) - Professional Notes

Topics Covered

Main Focus: Kafka Consumer Groups, Rebalancing, and Partition Assignment Strategies.

Questions and Professional Interview-Style Answers

Q1. What is a Kafka Consumer Group?

- A Kafka consumer group is a set of consumers working together to consume data from topics.
- Each partition of a topic is consumed by only one consumer within a group at any given time.
- Enables parallelism and fault tolerance.

Production Tip: If a consumer crashes, another consumer in the same group will take over its partition.

Q2. What triggers a Kafka Rebalance?

- Joining or leaving of a consumer in the group.
- New partitions added to a subscribed topic.
- Consumer heartbeat failures (e.g., session timeout).
- Coordinator failures.

Production Issue Example: In production, rebalances often spike lag temporarily; tuning session timeout reduces unnecessary rebalances.

Q3. What happens during a Rebalance?

- Consumers stop consuming.
- Group Coordinator triggers re-partitioning.
- Partitions are reassigned among consumers.
- Consumers rejoin the group with new assignments.

Q4. What are common problems caused by frequent Rebalancing?

- Temporary lag buildup.
- Processing delays.
- Increased GC (Garbage Collection) pauses.
- Resource wastage due to consumer reinitialization.

Q5. How do session.timeout.ms and heartbeat.interval.ms affect Rebalancing?

- `session.timeout.ms`: How long broker waits before considering a consumer dead.
- `heartbeat.interval.ms`: How frequently a consumer sends heartbeat to broker.
- **Tuning Needed:** Heartbeat interval should be significantly smaller than session timeout (typically 1/3).

Trade-Off: Larger session timeout reduces unnecessary rebalances but slower in detecting genuine failures.

Q6. Explain RangeAssignor.

- Assigns consecutive partitions to consumers.

- Can lead to partition skew (first consumers get more partitions).

Example: Partition 0,1,2 to Consumer1; 3,4,5 to Consumer2.

Q7. Explain RoundRobinAssignor.

- Partitions are assigned to consumers one by one in a round-robin manner.
- Balances partitions better if topic subscription is same for all consumers.

Example: Consumer1 gets Partition 0,2,4; Consumer2 gets 1,3,5.

Q8. Explain StickyAssignor.

- Attempts to retain previous partition assignments as much as possible during rebalancing.
- Reduces data processing disruption.

Good for: High-throughput production systems needing minimal lag during rebalancing.

Q9. What is CooperativeStickyAssignor?

- Introduces cooperative rebalancing: consumers can incrementally adjust partition assignment without full stop-the-world rebalance.
- Reduces downtime and lag.

Production Tip: Default recommended assignor for large-scale production systems since Kafka 2.4+.

Q10. What production issues can happen during rebalancing?

- Increased Consumer Lag.
- Temporary message duplication or ordering delays.

- Timeouts if rebalancing takes too long.
- Load spikes on brokers.

Q11. What are signs of rebalance storms?

- Frequent consumer group re-joins.
- Lag graphs showing periodic sharp spikes.
- Group coordinator logs showing rejoin/failures every few minutes.

Q12. What tuning helps in avoiding rebalance storms?

- Increase `session.timeout.ms` moderately.
- Tune `max.poll.interval.ms` if poll times are high.
- Use `CooperativeStickyAssignor`.
- Optimize application processing time to avoid heartbeats missing.

Q13. What happens if one consumer is slow in a group?

- Causes rebalancing if session timeout occurs.
- Can impact throughput of the entire group.
- May need to partition reassign or vertically scale consumers.

Q14. How would you monitor and detect unhealthy consumer groups?

- Metrics:
  - Consumer Lag (group.lag)

    - Under-Rebalanced Groups (frequent rebalances)
    - Heartbeat Misses
- Tools:
    - Burrow
    - Cruise Control
    - Grafana + Prometheus

Action: Set lag and rejoin frequency alerts in production.

End of Day 2 Notes

Ready for Day 3: Partition Assignment Strategies deep dive.