## Problem Statement

1. ARCV Receiver stoped consuming message from Kafka topic even new message was produced into assigned topic partition

2. Zookeeper didn't contains any offset information of Receiver Spout

3. After resubmitted topology, ARCV Receiver consumed message as expectation in both individual functional test or performance test can be performed

4. After mass inputs testing (e.g. 1000+ performance test), ARCV Receiver stops consuming message in individual functional test

## Why Did This Happen

1. Tuples were not acknowledged properly in following bolts of topology

2. Spout didn’t commit offset for polled records since tuples were not acknowledged

3. The number of polled offsets (records) reach the limit (1000), spout stopped polling message from topic. This limit is defined in property “max.uncommitted.offsets”, and if limit is reached, no more offsets can be polled until the next successful commit sets the number of pending offsets below the threshold.

4. After resubmitted topology the state “numUncommittedOffsets” was reset to 0, thus spout was able to poll messages

## Solutions

1. Without redeployment: Resubmit topology every time issue happens, clearance of messages inside topics are required as well.

2. With redeployment: Since latest version 1.21.0 and 1.22.0, all the tuples are carefully acknowledged in all bolts, spout will commit offset and release the pending offsets

## More about KafkaSpout

There are some properties control the performance of KafkaSpout:

* offset.commit.period.ms : specifies the period of time (in milliseconds) after which the spout commits to Kafka.
  + When the spout is committing offset, it is not fetching new records nor processing new tuples.
  + The lower value this parameter is, the more often spout commits offset, the less throughput can spout achieve
* max.uncommitted.offsets: defines the maximum number of polled offsets (records) that can be pending commit before another poll can take place
  + The higher this parameter is, the more memory consumed by Spout. Each offset uses eight bytes of memory, which means that a value of 10000000 (10MB) uses about 80MB of memory
* poll.timeout.ms : specifies the time (in milliseconds) spent polling if data is not available