Kafka-２

Kafka High Available机制

* Data Replication

启用原因：若无 Replication，一旦某一个 Broker 宕机，则其上所有的 Partition 数据都不可被消费，这与 Kafka 数据持久性及 Delivery Guarantee 的设计目标相悖。同时 Producer 都不能再将数据存于这些 Partition 中。

工作机制：消息被发布到某个 Partition 时，先通过 ZooKeeper 找到该 Partition 的 Leader，并发送给该Leader。Leader 会将该消息写入其本地 Log。每个 Follower 都从 Leader pull 数据。每个 Follower 在接收到数据后就立马向 Leader 发送 ACK，而非等到数据写入 Log 中。一旦 Leader 收到了 ISR 中的所有 Replica 的 ACK，该消息就被认为已经 commit 了，Leader 将增加 HW 并且向 Producer 发送 ACK。

ISR：Kafka 在 ZooKeeper 中动态维护了一个 ISR（in-sync replicas），这个 ISR 里的所有 Replica 都跟上了Leader最新的数据。

* Leader Election

同一个 Partition 可能会有多个 Replica，需要在其之间选出一个 Leader，Producer 和 Consumer 只与这个 Leader 交互，其它 Replica 作为 Follower 从 Leader 中复制数据。

只有 ISR 里的成员才有被选为 Leader 的可能。

所有 Partition 的 Leader 选举都由从所有 broker 中选出的controller 决定。controller 会将 Leader 的改变直接通过 RPC 的方式通知需为为此作为响应的 Broker。同时 controller 也负责增删 Topic 以及 Replica 的重新分配。

Replication 工具

* Topic Tool

$KAFKA\_HOME/bin/kafka-topics.sh

用于列出所有 Topic，及创建、删除、修改、查看某个 Topic。

* Replica Verification Tool

$KAFKA\_HOME/bin/kafka-replica-verification.sh

用于验证所指定的一个或多个 Topic 下每个 Partition 对应的所有 Replica 是否都同步。

* Preferred Replica Leader Election Tool

$KAFKA\_HOME/bin/kafka-preferred-replica-election.sh --zookeeper localhost:2181

用于在AR 范围内调整Broker 间 Leader 分配的平衡

AR（Assigned Replicas）：某个 Partition 的 Replica 列表叫作AR。AR 中的第一个 Replica 即为“Preferred Replica”。

* Kafka Reassign Partitions Tool

$KAFKA\_HOME/bin/kafka-reassign-partitions.sh --zookeeper localhost:2181

用于调整 Partition 的 AR 位置及数量（即改变该 Topic 的 replication factor）

有三种使用模式：

* + - generate 模式，给定需要重新分配的 Topic，自动生成 reassign plan（并不执行）
  + execute 模式，根据指定的 reassign plan 重新分配 Partition
  + verify 模式，验证重新分配 Partition 是否成功
* State Change Log Merge Tool

$KAFKA\_HOME/bin/kafka-run-class.sh kafka.tools.StateChangeLogMerger

用于从整个集群的 Broker 上收集状态改变日志，并生成一个集中的格式化的日志以帮助诊断状态改变相关的故障。

心得：

在实现了系统逻辑后，系统的可用性是长久使用的保证，在设计时需要考虑。

参考：

<https://www.infoq.cn/article/kafka-analysis-part-2>

<https://www.infoq.cn/article/kafka-analysis-part-3>

<https://zhuanlan.zhihu.com/p/30270149>

<https://segmentfault.com/a/1190000014918849>

次回：

<https://blog.csdn.net/high2011/article/details/79526652>