

大胜.彼得:20:38:15

实际应用中，kafka 集群必须要依赖 zk 吗

答：需要,没与 zk,kafka 就启动不了

攀登:20:38:22

you

大胜.彼得:20:38:38

明白了

杜庆奎:20:39:27

nacos 能用吗

黄欣健:20:40:44

论资排辈

任献良:20:42:18

按照 broker id?

湮汐:20:42:33

谁先连上选谁

大胜.彼得:20:42:45

就用原来的 zk 选举算法吗

姚远:20:42:46

临时节点 第一个注册

任献良:20:42:49

配置文件里面的 id

攀登:20:42:50

id 排序，小的为 controller

马德成:20:43:04

投票

攀登:20:44:23

zookeeper 应用 master 选举

杜庆奎:20:57:48

一部分人先富起来，消灭富不起来的

湮汐:20:58:04

broker 什么时候响应 producer 消息提交成功？

答：下面会讲到消息发送的可靠性机制。

大胜.彼得:21:07:02

数据同步 是由 zk 完成的吗

soso-李卓:21:07:34

新 leader 的数据比 follower 少，所以 follow 要截断

soso-李卓:21:07:59

返回 producer 失败

吕伟明:21:08:53

不是按照消息数量来吗，如果 c 是 leader 了 5 的消息不会丢了吗、

soso-李卓:21:09:07

我也是这意思

余爽:21:10:55

6

姚远:21:10:56

6 不一样了

任献良:21:11:01

6 数据不一致

soso-李卓:21:12:04

从 hw 开始截断

黄欣健:21:13:23

数据丢失

各三名:21:13:36

数据丢了

任献良:21:13:38

丢了一条 A 的 6

黄欣健:21:13:38

原来没同步的 6

各三名:21:13:43

A 的

soso-李卓:21:14:37

producer 知道么？

杨海波:21:14:49

A 重新启动后，从哪得知宕机时的 HW？

Free3610:21:14:49

老雷，不是同步完后才回复 ack 吗

姚远:21:15:02

producer 看设置。设置 ack 是 all 就不算丢吧

Ghost:21:15:19

丢了咋办？？？？反正也是不重要的数据 丢就丢了？

大胜.彼得:21:16:12

截断后再同步，6 不是同步过去了吗？为啥说 6 还丢失

黄欣健:21:16:17

不过雷哥开始的时候就说了，Kafka 是为了海量日志而生的，丢失数据也莫得办法

答：对的,kafka 没办法保证所有的数据都不丢。

吕伟明:21:17:10

但是选举 leader 很随意啊，论资排辈？为啥不按消息数量，这样还能少丢点

答：大多数情况下,消息数量都是差不多的,不会说差别非常大

吕伟明:21:18:32

最后一台同步的速度

黄欣健:21:18:36

同步速度

湮汐:21:18:36

差的

攀登:21:18:37

差

大胜.彼得:21:18:37

差的

任献良:21:18:38

差

黄欣健:21:18:39

查的

各三名:21:18:43

数据最少的那个

攀登:21:18:47

短板效应

Free3610:21:18:48

差的

soso-李卓:21:19:30

多丢一条数据

攀登:21:19:31

5 丢失

杨海波:21:20:25

A 重新启动后，从哪得知宕机时的 HW？

答：因为它自己保存了 HW.

湮汐:21:20:26

论资排辈 只会选择 ISR 里面的 follower

大胜.彼得:21:20:45

之前重叠的 2 个 6 数据是不是不一样

答：是的,所依导致 6 数据丢失了.

大胜.彼得:21:20:54

所以说 6 丢失了

各三名:21:21:10

刚刚说按顺序来 按什么顺序来？

答：按 ISR 顺序.

大胜.彼得:21:21:19

明白了

任献良:21:24:04

黄欣健:21:24:21

不过 Leader 恢复的时候，咋知道自己之前宕机过的，雷哥

答: 因为它自己保存了 HW.

小孟-北京-4 年:21:25:17

起来的时候回连接 zk, 连接上 zk, 不就知道自己是不是 leader 了

各三名:21:25:53

如果正常 HW 和 LEO 应该是一样的 不一样说明宕机过 个人猜测

答: 正常情况下, HW 和 LEO 不一定是一样的, LEO 各是各的, HW 是大家的

soso-李卓:21:26:18

数据丢失会在哪里提现出来? 还是就无从可查?

答: 前面演示的动画里, 6 号数据丢失就是.

攀登:21:26:20

Leader 宕机是 broker 集群 异常?

余爽:21:27:05

宕机恢复后从 HW 开始同步, 而不是 LEO

答: 对的

大胜.彼得:21:27:20

不一样不一定宕机, 同步慢了也不一样

任献良:21:27:38

LEO 会移动到 HW 位置

soso-李卓:21:28:05

是我们能知道丢失了 6 号么？还是根本不知道 6 号曾经来过？

答：是 kafka 里边根本不知道有 6 号数据。LeaderA 里面有 6 数据，但是 B 和 C 还没同步到 6 数据，这时候 A 宕机了，B 成为 Leader，对 B 和 C 来说 6 相当于没来过。

soso-李卓:21:28:32

服务端查不到 6 号是吧？

黄欣健:21:30:13

Follower 同步时是会读取 Leader 的 LEO 值吗，然后就知道自己应该同步到多少结束

答：不用读 Leader 的 LEO，直接是同步到最后一个。

soso-李卓:21:30:48

没有写个错误日志一类的东西 记录 6 号 是吧？

小孟-北京-4 年:21:32:13

leo 实时变化，读它没有意义

soso-李卓:21:32:16

设计目标，不是为了精确保存 所以丢就丢把 是吧 哈

黄欣健:21:32:52

那 LEO 感觉没啥用呀。对外能够消费的只是 HW 那条线以

上的消息，宕机后同步也是从 HW 开始同步

soso-李卓:21:33:30

是的 leo 到 hw 东西 理论可以记住

姚远:21:34:36

雷老师 我请教一下，一个已经线上运行很久的 topic ，我新增一个新的消费 group，不指定 offset 的情况下 是从哪开始消费。

大胜.彼得:21:34:49

后面还有哪些课程和 zk 有关系的？

黄欣健:21:36:41

我觉得 应该是从你消费 group 的那个配置属性 auto.offset.reset 这里开始消费吧

黄欣健:21:36:46

我猜的，嘻嘻

姚远:21:39:32

我觉得 也是可以设置的。。。

攀登:21:54:40

存在

余爽:21:54:43

存在

余爽:21:54:53

HW 截断

黄欣健:21:56:37

能

马崇:21:56:37

可以

湮汐:21:56:40

能

蒋伟:21:56:58

不能

大胜.彼得:21:57:14

能，但不一定同步

任献良:21:57:15

能

马德成:21:57:18

能

杨升玮-上海:21:57:19

能

任献良:21:57:23

服务器的问题 不是生产出的问题

各三名:21:57:30

不能

攀登:21:58:17

加一个 一定 吧

黄欣健:21:58:20

为啥，leader 不是响应了 ack 吗

答：响应 ACK 了,消息还没同步就挂了.

杨腾飞:21:58:55

HW

黄欣健:21:58:58

哦哦

任献良:21:59:00

只有可以消费 才算成功

攀登:21:59:01

消息因为内部异常，而丢失了。

各三名:21:59:01

这个问题的关键是什么叫 发送成功

攀登:21:59:23

可以被消费

余爽:22:00:21

不能

大胜.彼得:22:00:22

不能

攀登:22:00:22

不能

soso-李卓:22:00:23

No

马德成:22:00:26

能

杜庆奎:22:00:26

可以

任献良:22:00:28

也不能

吕伟明:22:00:32

能

杨海波:22:00:32

不能

黄欣健:22:00:34

可以把，能消费就成，不能消费就失败

xxzx_3910730:22:00:35

能

Free3610:22:00:38

可以

soso-李卓:22:00:44

能

杜庆奎:22:00:54

能

湮汐:22:00:54

响应成功未必成，响应失败一定是失败

攀登:22:00:58

消息没有被写入 broker

任献良:22:01:02

没有 ack 可以重发

各三名:22:01:04

出现 HW 截断时 不能

大胜.彼得:22:01:07

不能

soso-李卓:22:01:12

反过来说就很容易明白了，能的

黄欣健:22:01:23

没有 ack 就是失败

大胜.彼得:22:01:44

收到 ACK，也可能失败的

攀登:22:01:51

但是, 收到 ack,也有可能失败的。

杨腾飞:22:01:57

还有其他失败情况他不一定知道

任献良:22:01:59

没有收到一定失败

黄欣健:22:02:01

被雷哥带沟里了

攀登:22:02:45

成功是不确定的, 失败是确定的。

小孟-北京-4 年:22:04:25

01-1 在性能上差多少?

答: 这个不确定的,跟实际的环境有关.

soso-李卓:22:04:40

这个配置是全局性的么?

答: 这是 producer 的配置,不是 kafka 的.

湮汐:22:05:16

producer 的配置

soso-李卓:22:05:38

哦 明白了

黄欣健:22:05:40

雷哥，如果一开始往 A （leader）发送消息 1 2 3，然后 B 同步完 1 的时候，A 宕机了，然后 B 当选 leader，这时候重发消息，还是重发 1 2 3 吗

黄欣健:22:06:22

对

soso-李卓:22:06:28

看配置啊

攀登:22:06:38

1 2 3

黄欣健:22:06:53

我继续发 1 2 3

soso-李卓:22:06:53

—1:1 2 3

黄欣健:22:07:14

那 B 接受到消息会产生消息重复吧，两个 1

soso-李卓:22:07:32

0:不重发

soso-李卓:22:07:48

1: 2 3 ?

任献良:22:08:57

不是一条回复一个 ack 吗？

任献良:22:09:53

哦了

余爽:22:09:59

挂了之后不是就不在 ISR 列表里了吗？这种情况也不算成功吗？

吕伟明:22:10:10

全量同步吗

soso-李卓:22:10:25

那刚刚说的 0 的那种直接返回 是啥意思？ 也要等都同步完才返回么？

杜庆奎:22:10:55

A 回复后之前接收的消息会丢失吗？还会消费吗

杜庆奎:22:11:01

恢复

攀登:22:12:01

0 类似 UDP

soso-李卓:22:12:31

那刚刚说的 1 的那种直接返回 ？ 也要等都同步完才返回么？

答: 1 的那种只要 Leader 收到了就直接回复了.

大胜.彼得:22:12:35

A 恢复

杜庆奎:22:12:47

假如 A 宕机恢复后

任献良:22:13:01

没好好听课

黄欣健:22:13:09

01-1 标准越来越严格

答: 总结的很到位

soso-李卓:22:13:37

有问题, 如果是客户端配置

答: 这本来就是客户端配置的,配置的是 producer,不是 kafka

Free3610:22:14:19

重复提交了, rabbitMq 是可以保证的

soso-李卓:22:14:32

好的, 那怎么区分 leader follow ack?

攀登:22:15:33

pull 吧

黄欣健:22:16:14

poll(timeout)

黄欣健:22:27:11

集群没了

湮汐:22:27:44

其实自己有毛病

任献良:22:28:02

一般活不过 2 集

soso-李卓:22:30:36

总体意思是，最后选中了那个坏的节点，导致把好的都丢弃了

soso-李卓:22:36:28

超时呗

湮汐:22:39:01

那自动提交有啥好处呢？感觉只会带来麻烦

任献良:22:39:01

那不就死循环了 一直消费这 500 条

任献良:22:40:07

买两台性能好的机器

黄欣健:22:40:10

改为同异手动提交

soso-李卓:22:40:23

抖动咋办？解决不彻底啊

杨腾飞:22:40:43

代码记录 offset，超市先重置 offset 呗

soso-李卓:22:40:47

手动可控

大胜.彼得:22:41:22

还是手动可靠

谢尊锦:22:41:56

手动会不会网络开销大

马崇:22:44:38

消息消费后，消息是还在的吗

soso-李卓:22:44:59

一直在文件里啊

大胜.彼得:22:46:00

缩短提交时间

湮汐:22:46:19

手动提交也没办法完全解决吧，只能消费方做幂等

黄欣健:22:46:21

那其实还要自动提交干嘛手动也就加一行代码

soso-李卓:22:46:34

手动也会重复啊

攀登:22:46:53

调整 offset 更新机制

攀登:22:47:11

消费完才更新 offset

soso-李卓:22:47:34

幂等 是消费端考虑的问题

答: 对的

soso-李卓:22:48:06

只有一个 consumer

马崇:22:48:50

这些 consumer 是指一个组的吗

答: 不同组的不存在重复消费

黄欣健:22:49:18

幂等是啥, 好像没听过

soso-李卓:22:49:27

ooo

Free3610:22:49:58

请求多少次都不影响

Free3610:22:50:06

提交

王腾龙:22:50:14

业务幂等

soso-李卓:22:50:15

离散数学

Free3610:22:50:17

CAP 理论

黄欣健:22:50:24

哦哦

soso-李卓:22:51:03

producer 同时接受 leader 和 follower 的 ack? 1 和 -1 对 producer 接受返回值有什么区别?

答: 只有 leader 才会 ack, leader 和 follower 是主备关系. 对于 producer 来说没有区别, 但是对于 kafka 来说, 1 的时候收到 ack 了但是没成功, 就会丢失消息, 但是对于 -1 来说, 收到 ack 了一定是成功过的.

黄欣健:22:51:57

刚有个兄弟问, 如果有一个 topic 已经存在很久, 然后新加入了一个新的消费组, 会从哪里开始消费

答 : 这个 是 可 控 的

```
// earliest: 指定offset为第一条offset  
// latest: 指定offset为最后一条offset  
properties.put("auto.offset.reset", "earliest");|
```

马崇:22:52:12

主备

湮汐:22:52:18

健哥你这个问题 看配置

soso-李卓:22:52:19

主呗

黄欣健:22:52:48

我也是觉得看配置，看 `auto.offset.reset`

湮汐:22:52:58

对的 就是这个配置

soso-李卓:22:53:55

所以 问题的第二部分，1 和 -1 对 producer 接受返回值有什么区别？

soso-李卓:22:54:53

那她咋判断啊，没区别，客户端配置

soso-李卓:22:55:24

是否重发

soso-李卓:22:56:29

好的，我再考虑下