## kafka基础运维

1. 分两种日志 数据日志 操作日志
   * + 1. 数据日志是指Kafka的topic中存储的数据，这种日志的路径是在$KAFKA\_HOME/config/server.properties文件中配置，配置项为log.dirs。如果此项没有被配置，默认会使用配置项 log.dir（请仔细观察，两个配置项最后查了一个s）。log.dir的默认路径为/tmp/kafka-logs，大家知道，/tmp路径下的文件在计算机重启的时候是会被删除的，因此，强烈推荐将文件目录设置在其他可以永久保存的路径。另一种日志是操作日志，类似于我们在自己开发的程序中输出的log日志（log4j），这种日志的路径是在启动Kafka的路径下。比如一般我们在$KAFKA\_HOME路径下启动Kafka服务，那么操作日志的路径为$KAFKA\_HOME/logs。

2、数据日志清理两种方式

日志被发布的时间删除

Kafka提供了配置项让我们可以按照日志被发布的时间来删除。它们分别是：

log.retention.ms

log.retention.minutes

log.retention.hours

log.retention.ms表示日志会被保留多少毫秒，如果为null，则Kafka会使用使用log.retention.minutes配置项。log.retention.minutes表示日志会保留多少分钟，如果为null，则Kafka会使用log.retention.hours选项。默认情况下，log.retention.ms和log.retention.minutes均为null，log.retention.hours为168，即Kafka的数据日志默认会被保留7天。如果想修改Kafka中数据日志被保留的时间长度，可以通过修改这三个选项来实现。

日志文件的size删除

按size删除

Kafka除了提供了按时间删除的配置项外，也提供了按照日志文件的size来删除的配置项：

log.retention.bytes

即日志文件到达多少byte后再删除日志文件。默认为-1，即无限制。需要注意的是，这个选项的值如果小于segment文件大小的话是不起作用的。segment文件的大小取决于log.segment.bytes配置项，默认为1G。

另外，Kafka的日志删除策略并不是非常严格的（比如如果log.retention.bytes设置了10G的话，并不是超过10G的部分就会立刻删除，只是被标记为待删除，Kafka会在恰当的时候再真正删除），所以请预留足够的磁盘空间。当磁盘空间剩余量为0时，Kafka服务会被kill掉。

因为分区工具接受将topic列表写入json文件的形式，所以你需要指定需要移动的topics并创建json文件，就像下面这样：

cat topics-to-move.json

{ “topics”: [{“topic”: “foo1”}, {“topic”: “foo2”}], “version”:1 }

当json文件准备好 了以后，使用重新分区工具自动生成一个候选的重新分配计划：

bin/kafka-reassign-partitions.sh –zookeeper localhost:2181 –topics-to-move-json-file topics-to-move.json –broker-list “5,6” –generate

Current partition replica assignment

{“version”:1,

“partitions”:[{“topic”:”foo1”,”partition”:2,”replicas”:[1,2]},

{“topic”:”foo1”,”partition”:0,”replicas”:[3,4]},

{“topic”:”foo2”,”partition”:2,”replicas”:[1,2]},

{“topic”:”foo2”,”partition”:0,”replicas”:[3,4]},

{“topic”:”foo1”,”partition”:1,”replicas”:[2,3]},

{“topic”:”foo2”,”partition”:1,”replicas”:[2,3]}]

}

Proposed partition reassignment configuration

{“version”:1,

“partitions”:[{“topic”:”foo1”,”partition”:2,”replicas”:[5,6]},

{“topic”:”foo1”,”partition”:0,”replicas”:[5,6]},

{“topic”:”foo2”,”partition”:2,”replicas”:[5,6]},

{“topic”:”foo2”,”partition”:0,”replicas”:[5,6]},

{“topic”:”foo1”,”partition”:1,”replicas”:[5,6]},

{“topic”:”foo2”,”partition”:1,”replicas”:[5,6]}]

}

分区工具生成的候选计划将会将topic foo1,foo2的所有分区移动到broker 5,6。记住，尽管到了这一步，partition的移动却尚未开始，而仅仅告诉你当前的分区情况（Current partition replica assignment）和建议的新分区方案（Proposed partition reassignment configuration）。当前分区情况应该保留起来，以防你想要回滚。新分区方案应该被存入一个josn文件中，（比如：expand-cluster-reassignment.json），然后作为分区工具 –execute选项的输入文件。就像下面这样：

bin/kafka-reassign-partitions.sh –zookeeper localhost:2181 –reassignment-json-file expand-cluster-reassignment.json –execute

Current partition replica assignment

{“version”:1,

“partitions”:[{“topic”:”foo1”,”partition”:2,”replicas”:[1,2]},

{“topic”:”foo1”,”partition”:0,”replicas”:[3,4]},

{“topic”:”foo2”,”partition”:2,”replicas”:[1,2]},

{“topic”:”foo2”,”partition”:0,”replicas”:[3,4]},

{“topic”:”foo1”,”partition”:1,”replicas”:[2,3]},

{“topic”:”foo2”,”partition”:1,”replicas”:[2,3]}]

}

Save this to use as the –reassignment-json-file option during rollback

Successfully started reassignment of partitions

{“version”:1,

“partitions”:[{“topic”:”foo1”,”partition”:2,”replicas”:[5,6]},

{“topic”:”foo1”,”partition”:0,”replicas”:[5,6]},

{“topic”:”foo2”,”partition”:2,”replicas”:[5,6]},

{“topic”:”foo2”,”partition”:0,”replicas”:[5,6]},

{“topic”:”foo1”,”partition”:1,”replicas”:[5,6]},

{“topic”:”foo2”,”partition”:1,”replicas”:[5,6]}]

}

最后，–verify选项可以用来检查重新分配分区的状态。注意此处应该使用与执行 –execute 选项时相同的expand-cluster-reassignment.json 文件：

bin/kafka-reassign-partitions.sh –zookeeper localhost:2181 –reassignment-json-file expand-cluster-reassignment.json –verify

Status of partition reassignment:

Reassignment of partition [foo1,0] completed successfully

Reassignment of partition [foo1,1] is in progress

Reassignment of partition [foo1,2] is in progress

Reassignment of partition [foo2,0] completed successfully

Reassignment of partition [foo2,1] completed successfully

Reassignment of partition [foo2,2] completed successfully

自定义分区分配和移动

分区重新分配工具也可以被用来移动broker被选中的分区。当使用这种方式的时候，我们假设用户了解重新分配分区方案并且不需要使用分区工具去自动生成候选重新分区方案。为了高效，我们跳过了–generate步骤，直接来到–execute步骤

举例来说，下面的例子将topic foo1的partition 0移动到 broker 5,6，将topic foo2的partition 1 移动到broker 2,3：

第一步来手写自定义重新分区计划的json文件：

cat custom-reassignment.json

{“version”:1,”partitions”:[{“topic”:”foo1”,”partition”:0,”replicas”:[5,6]},{“topic”:”foo2”,”partition”:1,”replicas”:[2,3]}]}

然后，使用–execute选项和json文件来开始重新分配分区的操作：

bin/kafka-reassign-partitions.sh –zookeeper localhost:2181 –reassignment-json-file custom-reassignment.json –execute

Current partition replica assignment

{“version”:1,

“partitions”:[{“topic”:”foo1”,”partition”:0,”replicas”:[1,2]},

{“topic”:”foo2”,”partition”:1,”replicas”:[3,4]}]

}

Save this to use as the –reassignment-json-file option during rollback

Successfully started reassignment of partitions

{“version”:1,

“partitions”:[{“topic”:”foo1”,”partition”:0,”replicas”:[5,6]},

{“topic”:”foo2”,”partition”:1,”replicas”:[2,3]}]

}

用–verify选项来查看被重新分配分区的状态。注意此处应该使用与执行 –execute 选项时相同的expand-cluster-reassignment.json 文件：

bin/kafka-reassign-partitions.sh –zookeeper localhost:2181 –reassignment-json-file custom-reassignment.json –verify

Status of partition reassignment:

Reassignment of partition [foo1,0] completed successfully

Reassignment of partition [foo2,1] completed successfully