

Kafka管理工具介绍

Kafka内部提供了许多管理脚本，这些脚本都放在\$KAFKA_HOME/bin目录下，而这些类的实现都是放在源码的kafka/core/src/main/scala/kafka/tools/路径下。

Consumer Offset Checker

Consumer Offset Checker主要是运行kafka.tools.ConsumerOffsetChecker类，对应的脚本是kafka-consumer-offset-checker.sh，会显示出Consumer的Group、Topic、分区ID、分区对应已经消费的Offset、logSize大小，Lag以及Owner等信息。

如果运行kafka-consumer-offset-checker.sh脚本的时候什么信息都不输入，那么会显示以下信息：

```
[iteblog@www.iteblog.com /]$ bin/kafka-consumer-offset-checker.sh
Check the offset of your consumers.
Option                Description
-----
--broker-info          Print broker info
--group                Consumer group.
--help                 Print this message.
--retry.backoff.ms <Integer>  Retry back-off to use for failed
                             offset queries. (default: 3000)
--socket.timeout.ms <Integer> Socket timeout to use when querying
                             for offsets. (default: 6000)
--topic                Comma-separated list of consumer
                             topics (all topics if absent).
--zookeeper            ZooKeeper connect string. (default:
                             localhost:2181)
```

我们根据提示，输入的命令如下：

```
[iteblog@www.iteblog.com /]$ bin/kafka-consumer-offset-
checker.sh --zookeeper www.iteblog.com:2181 --topic test --group spark --broker-info
Group    Topic  Pid Offset    logSize    Lag    Owner
spark    test   0  34666914    34674392    7478    none
spark    test   1  34670481    34678029    7548    none
spark    test   2  34670547    34678002    7455    none
spark    test   3  34664512    34671961    7449    none
spark    test   4  34680143    34687562    7419    none
```

```
spark test 5 34672309 34679823 7514 none
spark test 6 34674660 34682220 7560 none
```

BROKER INFO

2 -> www.iteblog.com:9092

5 -> www.iteblog.com:9093

4 -> www.iteblog.com:9094

7 -> www.iteblog.com:9095

1 -> www.iteblog.com:9096

3 -> www.iteblog.com:9097

6 -> www.iteblog.com:9098

Dump Log Segment

有时候我们需要验证日志索引是否正确，或者仅仅想从log文件中直接打印消息，我们可以使用kafka.tools.DumpLogSegments类来实现，先来看看它需要的参数：

```
[iteblog@www.iteblog.com /]$ bin/kafka-run-class.sh kafka.tools.DumpLogSegments
```

Parse a log file and dump its contents to the console, useful for debugging a seemingly corrupt log segment.

Option	Description
-----	-----
--deep-iteration	if set, uses deep instead of shallow iteration
--files <file1, file2, ...>	REQUIRED: The comma separated list of data and index log files to be dumped
--key-decoder-class	if set, used to deserialize the keys. This class should implement kafka.serializer.Decoder trait. Custom jar should be available in kafka/libs directory. (default: kafka.serializer.StringDecoder)
--max-message-size <Integer: size>	Size of largest message. (default: 5242880)
--print-data-log	if set, printing the messages content when dumping data logs
--value-decoder-class	if set, used to deserialize the messages. This class should implement kafka.serializer.Decoder trait. Custom jar should be available in kafka/libs directory. (default: kafka.serializer.StringDecoder)
--verify-index-only	if set, just verify the index log

without printing its content

很明显，我们在使用kafka.tools.DumpLogSegments的时候必须输入--files，这个参数指的就是Kafka中Topic分区所在的绝对路径。分区所在的目录由config/server.properties文件中log.dirs参数决定。比如我们想看/home/q/kafka/kafka_2.10-0.8.2.1/data/test-4/00000000000034245135.log日志文件的相关情况可以使用下面的命令：

```
[iteblog@www.iteblog.com /]$ bin/kafka-run-class.sh kafka.tools.DumpLogSegments --files /iteblog/data/test-4/00000000000034245135.log
Dumping /home/q/kafka/kafka_2.10-0.8.2.1/data/test-4/00000000000034245135.log
Starting offset: 34245135
offset: 34245135 position: 0 invalid: true payloadsize: 4213 magic: 0 compresscodec: NoCompressionCodec crc: 865449274 keysize: 4213
offset: 34245136 position: 8452 invalid: true payloadsize: 4657 magic: 0 compresscodec: NoCompressionCodec crc: 4123037760 keysize: 4657
offset: 34245137 position: 17792 invalid: true payloadsize: 3921 magic: 0 compresscodec: NoCompressionCodec crc: 541297511 keysize: 3921
offset: 34245138 position: 25660 invalid: true payloadsize: 2290 magic: 0 compresscodec: NoCompressionCodec crc: 1346104996 keysize: 2290
offset: 34245139 position: 30266 invalid: true payloadsize: 2284 magic: 0 compresscodec: NoCompressionCodec crc: 1930558677 keysize: 2284
offset: 34245140 position: 34860 invalid: true payloadsize: 268 magic: 0 compresscodec: NoCompressionCodec crc: 57847488 keysize: 268
offset: 34245141 position: 35422 invalid: true payloadsize: 263 magic: 0 compresscodec: NoCompressionCodec crc: 2964399224 keysize: 263
offset: 34245142 position: 35974 invalid: true payloadsize: 1875 magic: 0 compresscodec: NoCompressionCodec crc: 647039113 keysize: 1875
offset: 34245143 position: 39750 invalid: true payloadsize: 648 magic: 0 compresscodec: NoCompressionCodec crc: 865445580 keysize: 648
offset: 34245144 position: 41072 invalid: true payloadsize: 556 magic: 0 compresscodec: NoCompressionCodec crc: 1174686061 keysize: 556
offset: 34245145 position: 42210 invalid: true payloadsize: 4211 magic: 0 compresscodec: NoCompressionCodec crc: 3691302513 keysize: 4211
offset: 34245146 position: 50658 invalid: true payloadsize: 2299 magic: 0 compresscodec: NoCompressionCodec crc: 2367114411 keysize: 2299
offset: 34245147 position: 55282 invalid: true payloadsize: 642 magic: 0 compresscodec: NoCompressionCodec crc: 4122061921 keysize: 642
offset: 34245148 position: 56592 invalid: true payloadsize: 4211 magic: 0 compresscodec: NoCompressionCodec crc: 3257991653 keysize: 4211
offset: 34245149 position: 65040 invalid: true payloadsize: 2278 magic: 0 compresscodec: NoCompressionCodec crc: 2103489307 keysize: 2278
offset: 34245150 position: 69622 invalid: true payloadsize: 269 magic: 0 compresscodec: NoCompressionCodec crc: 792857391 keysize: 269
```

offset: 34245151 position: 70186 invalid: true payloadsize: 640 magic: 0 compresscodec: NoCompressionCodec crc: 791599616 keysiz: 640

可以看出，这个命令将Kafka中Message中Header的相关信息和偏移量都显示出来了，但是没有看到日志的内容，我们可以通过--print-data-log来设置。如果需要查看多个日志文件，可以以逗号分割。

导出Zookeeper中Group相关的偏移量

有时候我们需要导出某个Consumer group各个分区的偏移量，我们可以通过使用Kafka的kafka.tools.ExportZkOffsets类来满足。来看看这个类需要的参数：

```
[iteblog@www.iteblog.com /]$ bin/kafka-run-class.sh kafka.tools.ExportZkOffsets
```

Export consumer offsets to an output file.

Option	Description
-----	-----
--group	Consumer group.
--help	Print this message.
--output-file	Output file
--zkconnect	ZooKeeper connect string. (default: localhost:2181)

我们需要输入Consumer group，Zookeeper的地址以及保存文件路径：

```
[iteblog@www.iteblog.com /]$ bin/kafka-run-class.sh kafka.tools.ExportZkOffsets --group spark --zkconnect www.iteblog.com:2181 --output-file ~/offset
```

```
[iteblog@www.iteblog.com /]$ vim ~/offset
/consumers/spark/offsets/test/3:34846274
/consumers/spark/offsets/test/2:34852378
/consumers/spark/offsets/test/1:34852360
/consumers/spark/offsets/test/0:34848170
/consumers/spark/offsets/test/6:34857010
/consumers/spark/offsets/test/5:34854268
/consumers/spark/offsets/test/4:34861572
```

注意，--output-file参数必须在指定，否则会出错。

通过JMX获取metrics信息

我们可以通过kafka.tools.JmxTool类打印出Kafka相关的metrics信息。

```
[iteblog@www.iteblog.com /]$ bin/kafka-run-class.sh kafka.tools.JmxTool
```

Dump JMX values to standard output.

Option	Description
-----	-----
--attributes <name>	The whitelist of attributes to query. This is a comma-separated list. If no attributes are specified all objects will be queried.
--date-format <format>	The date format to use for formatting the time field. See java.text. SimpleDateFormat for options.
--help	Print usage information.
--jmx-url <service-url>	The url to connect to to poll JMX data. See Oracle javadoc for JMXServiceURL for details. (default: service:jmx:rmi:///jndi/rmi://: 9999/jmxrmi)
--object-name <name>	A JMX object name to use as a query. This can contain wild cards, and this option can be given multiple times to specify more than one query. If no objects are specified all objects will be queried.
--reporting-interval <Integer: ms>	Interval in MS with which to poll jmx stats. (default: 2000)

可以这么使用

```
[iteblog@www.iteblog.com /]$ bin/kafka-run-class.sh kafka.tools.JmxTool --jmx-  
url service:jmx:rmi:///jndi/rmi://www.iteblog.com:1099/jmxrmi
```

运行上面命令前提是在启动kafka集群的时候指定export
JMX_PORT=，这样才会开启JMX。然后就可以通过上面命令打印出Kafka所有的metrics信息。

Kafka数据迁移工具

这个工具主要有两个：kafka.tools.KafkaMigrationTool和kafka.tools.MirrorMaker。第一个主要是用于将Kafka 0.7上面的数据迁移到Kafka 0.8（<https://cwiki.apache.org/confluence/display/KAFKA/Migrating+from+0.7+to+0.8>）；而后者可以同步两个Kafka集群的数据（<https://cwiki.apache.org/confluence/pages/viewpage.action?pageId=27846330>）。都是从原端消费Messages，然后发布到目标端。

```
[iteblog@www.iteblog.com /]$ bin/kafka-run-class.sh kafka.tools.KafkaMigrationTool --kafka.07.jar kafka-0.7.19.jar --zkclient.01.jar zkclient-0.2.0.jar --num.producers 16 --consumer.config=sourceCluster2Consumer.config --producer.config=targetClusterProducer.config --whitelist=.*
```

```
[iteblog@www.iteblog.com /]$ bin/kafka-run-class.sh kafka.tools.MirrorMaker --consumer.config sourceCluster1Consumer.config --consumer.config sourceCluster2Consumer.config --num.streams 2 --producer.config targetClusterProducer.config --whitelist=".*"
```

日志重放工具

这个工具主要作用是从一个Kafka集群里面读取指定Topic的消息，并将这些消息发送到其他集群的指定topic中：

```
[iteblog@www.iteblog.com /]$ bin/kafka-replay-log-producer.sh
```

Missing required argument "[broker-list]"

Option	Description
--broker-list <hostname:port>	REQUIRED: the broker list must be specified.
--inputtopic <input-topic>	REQUIRED: The topic to consume from.
--messages <Integer: count>	The number of messages to send. (default: -1)
--outputtopic <output-topic>	REQUIRED: The topic to produce to
--property <producer properties>	A mechanism to pass properties in the form key=value to the producer. This allows the user to override producer properties that are not exposed by the existing command line arguments
--reporting-interval <Integer: size>	Interval at which to print progress info. (default: 5000)
--sync	If set message send requests to the brokers are synchronously, one at a time as they arrive.
--threads <Integer: threads>	Number of sending threads. (default: 1)

--zookeeper <zookeeper url> REQUIRED: The connection string for the zookeeper connection in the form host:port. Multiple URLs can be given to allow fail-over. (default: 127.0.0.1:2181)

Simple Consume脚本

kafka-simple-consumer-shell.sh工具主要是使用Simple Consumer API从指定Topic的分区读取数据并打印在终端：

```
bin/kafka-simple-consumer-shell.sh --broker-  
list www.iteblog.com:9092 --topic test --partition 0
```

更新Zookeeper中的偏移量

kafka.tools.UpdateOffsetsInZK工具可以更新Zookeeper中指定Topic所有分区的偏移量，可以指定成 earliest或者latest：

```
[iteblog@www.iteblog.com /]$ bin/kafka-run-class.sh kafka.tools.UpdateOffsetsInZK  
USAGE: kafka.tools.UpdateOffsetsInZK$ [earliest | latest] consumer.properties topic
```

需要指定是更新成earliest或者latest，consumer.properties文件的路径以及topic的名称

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接: **【】** ()