spark streaming 知识总结

日期：20170324

问题导读

1.DStreams 的含义是什么？
2.DStreams 提供哪两种类型的操作？
3.Transformations 操作分为哪两种类型？
4.本文说了哪些输入源？
5.什么是 batch？

RDD 与 job 之间的关系

Spark Streaming 是构建在 Spark 上的实时流计算框架，扩展了 Spark 流式大数据处理能力。Spark Streaming 将数据流以时间片为单位分割形成 RDD，使用 RDD 操作处理每一块数据，每块数据（也就是 RDD）都会生成一个 Spark Job 进行处理，最终以批处理的方式处理每个时间片的数据

*说明：*Spark 中的 Job 和 MR 中 Job 不一样不一样。MR 中 Job 主要是 Map 或者 Reduce Job。而 Spark 的 Job 其实很好区别，RDD 一个 action 算子就算一个 Job.

什么是 batch

Spark Streaming 生成新的 batch 并对它进行一些处理，每个 batch 中的数据都代表一个 RDD

理解 batch

间隔时间开始会创建，间隔时间内会积累

设置时间间隔的理解

我们知道 spark streaming 有个时间间隔。假如间隔为 1 秒，它是停下 1 秒，然后在接受 1 秒的数据，也就是说是间隔 1 秒，然后在接受数据，还是说接受 1 秒的数据。这里表面上没有太大的区别，其实在于理解的到不到位。说白了 batch 封装的是 1 秒的数据。

batch 创建

batch 在时间间隔开始被创建，在间隔时间内任何到达的数据都被添加到批数据中，间隔时间结束，batch 创建结束。

什么是 batch 间隔参数

间隔时间大小的参数被称之为 batch 间隔

batch 间隔范围一般为
500 毫秒到几分钟，由开发者定义。

spark streaming 应用

spark streaming 应用程序可以实时跟踪页面统计，训练机器学习模型或则自动检测异常，更多推荐参考让你真正明白 spark streaming
http://www.aboutyun.com/forum.php?mod=viewthread&tid=21141

DStreams 详解

DStreams 是 discretized streams 的缩写，是离散流的意思。
DStreams 是随着时间【推移】到达的一系列数据

每个 dstream 被表示为一个序列的 RDDS（因此名称"离散"）。
DStreams 可以不同的数据源创建，比如 flume，kafka，或则 hdfs.一旦构建，
DStreams 提供两种类型的操作：

transformations,产生一个新的 DStream
output operations，写数据到外部系统。
DStreams 提供许多与 RDD 相同的操作，外加一些关于时间的操作比如 slidingwindows【滑动窗口】。

DStreams 来源
1.外部数据源
2.通过 transformations 转换而来

Transformations 操作

分为有状态和无状态
Stateful transformations 需要 checkpointing，在 StreamingContext 中启用容错。
设置 checkpointing
ssc.checkpoint("hdfs://...")

Windowed transformations

window 操作需要两个参数，窗口持续时间和滑动持续时间。这两个必须是多个 StreamingContext 的 batch 时间区间。DStream 数据源时间间隔是 10 秒。想创建滑动窗口上一个 30 秒（或则上 3batches)），我们应该设置 windowDuration30 秒。sliding 时间间隔，默认是 batch 时间间隔，控制 DStream 刷新计算结果。如果我们的 DStream batch 时间区间为 10 秒，我们想计算我们的 window，只能在每个第二

batch。我们设置我们的 sliding 间隔为 20 秒。

输出操作【output operations】

保存 DStream 为文本文件【Scala】

[Scala] 纯文本查看 复制代码

?

```
1 ipAddressRequestCount.saveAsTextFiles("outputDir", "txt")
```

saveAsHadoopFiles()是 hadoop 输出格式，例如 Spark Streaming 没有 SaveAsSequenceFile()函数，我们可以保存为 SequenceFiles

Scala

[Scala] 纯文本查看 复制代码

?

```
1val writableIpAddressRequestCount = ipAddressRequestCount.map {
2(ip, count) => (new Text(ip), new LongWritable(count)) }
3writableIpAddressRequestCount.saveAsHadoopFiles[
4SequenceFileOutputFormat[Text, LongWritable]]("outputDir", "txt")
```

Java

[Java] 纯文本查看 复制代码

?

```
1JavaPairDStream<Text, LongWritable> writableDStream = ipDStream.mapToPair(
2new PairFunction<Tuple2<String, Long>, Text, LongWritable>() {
3public Tuple2<Text, LongWritable> call(Tuple2<String, Long> e) {
4return new Tuple2(new Text(e._1()), new LongWritable(e._2()));
5}});
6class OutFormat extends SequenceFileOutputFormat<Text, LongWritable> {};
7writableDStream.saveAsHadoopFiles(
8"outputDir", "txt", Text.class, LongWritable.class, OutFormat.class);
```

foreachRDD()

[Java] 纯文本查看 复制代码

?

```
1ipAddressRequestCount.foreachRDD { rdd =>
2rdd.foreachPartition { partition =>
3// Open connection to storage system (e.g. a database connection)
4partition.foreach { item =>
5// Use connection to push item to system
6}
7// Close connection
```

```
8}
9}
```

checkpointing 机制

spark streaming 主要机制 checkpointing,它将数据存储在一个可靠的文件系统，比如 hdfs.

checkpoint 的作用，用于恢复数据。它会定期保存状态到可靠的文件系统比如 hdfs,s3
比如你每 5-10 批数据设置 checkpointing。当发生丢失数据的时候，Spark Streaming 讲恢复最近的 checkpoint.随着 streaming application 的持续运行，checkpoint 数据占用的存储空间会不断变大。因此，需要小心设置 checkpoint 的时间间隔。设置得越小，checkpoint 次数会越多，占用空间会越大；如果设置越大，会导致恢复时丢失的数据和进度越多。一般推荐设置为 batch duration 的 5~10 倍。

输入源

spark streaming 支持多个数据源，一些核心的数据源，已被构建到 Streaming Maven artifact,其它可以通过额外的 artifact，比如 spark-streaming-kafka.
核心数据源比如 sockets，还有文件 和 Akka actors.

其它数据源

使用 kafka 必须引入 artifact：spark-streaming-kafka_2.10 到项目中。它提供 KafkaUtils 对象，通过 StreamingContext 和 JavaStreamingContext 创建 kafka 消息的 DStream.
因为它订阅多个 topic. DStream 创建由 topic 和 message 组成的对。我们可以调用 createStream()方法来创建 Stream。字符串分割开 ZooKeeper hosts, consumer group 的名称（唯一的名字），receiver 线程用于 topic.

Apache Kafka 订阅 Panda 的 topic【Scala】

[Scala] 纯文本查看 复制代码

```
?
1import org.apache.spark.streaming.kafka._
2...
3// Create a map of topics to number of receiver threads to use
4val topics = List(("pandas", 1), ("logs", 1)).toMap
5val topicLines = KafkaUtils.createStream(ssc, zkQuorum, group, topics)
6StreamingLogInput.processLines(topicLines.map(_._2))
```

Apache Kafka 订阅 to Panda's topic【Java】

[Java] 纯文本查看 复制代码

?

```
1 import org.apache.spark.streaming.kafka.*;
2 ...
3 // Create a map of topics to number of receiver threads to use
4 Map<String, Integer> topics = new HashMap<String, Integer>();
5 topics.put("pandas", 1);
6 topics.put("logs", 1);
7 JavaPairDStream<String, String> input =
8 KafkaUtils.createStream(jssc, zkQuorum, group, topics);
9 input.print();
```

推荐参照文章让你真正明白 spark streaming

http://www.aboutyun.com/forum.php?mod=viewthread&tid=21141

转载注明来自 about 云（www.aboutyun.com）

http://www.aboutyun.com/forum.php?mod=viewthread&tid=21307

更多 about 云文档

链接: https://pan.baidu.com/s/1c2EHU2O 密码: pr2y

搜索：

wwwaboutyuncom

qq7 群：552029443

捐助

hadoop 生态系统零基础入门及大数据实战