**Apache Kafka - Integration With Spark**

1. Giới thiệu Spark:

 Spark là hệ thống xử lý thực hiện phân tích dựa trên các mẩu dữ liệu nhỏ trên luồng dữ liệu. Spark cung cấp nhiều tính năng như SQL, Graph , Meachine Learning,…. Spark cho phép xây dựng và phân tích nhanh các mô hình dự đoán. Hơn nữa, nó còn cung cấp khả năng truy xuất toàn bộ dữ liệu cùng lúc, nhờ vậy ta không cần phải lấy mẫu dữ liệu.

1. Tại sao phải tích hợp Apache Kafka và Spark Streaming:  
   Người dùng có xu hướng tương tự trong việc sử dụng Apache Kafka và Spark Streaming. Do đó, việc tích hợp Apache Kafka và Spark ra đời. Mô hình này cho phép xây dựng và phân tích nhanh các mô hình dự đoán .Việc này làm tăng tính hiệu quả trong việc sử dụng Apache Kafka và đồng thời cung cấp đảm bảo độ tin cậy cao hơn.
2. Spark Streaming

Khi tích hợp với Apache Kafka , Spark cung cấp tính năng streaming, được dùng để xây dựng các mô hình real-time bằng cách nạp toàn bộ dữ liệu vào bộ nhớ.

Tính năng Spark Streaming nó sẽ nhận dữ liệu từ các nguồn streaming data, cụ thể ở đây là Kafka tạm gọi là receiver data. Sau đó, nó chia nhỏ cái receiver data và chứa trong các batch data, mỗi batch được xem như là một tập các RDD. Một dãy các RDD thì Spark gọi đó là DStream  
Spark Streaming có thể xuất các kết quả thành một chủ đề Kafka hoặc lưu trữ trong HDFS, cơ sở dữ liệu hoặc bảng điều khiển.  
SparkConf API  
Nó đại diện cho cấu hình cho một ứng dụng Spark. Được sử dụng để thiết lập các thông số Spark khác nhau làm cặp giá trị khóa.  
Lớp SparkConf có các phương pháp sau đây:  
• set(string key, string value) - thiết lập biến cấu hình.  
• remove(string key) - xóa một key từ cấu hình.  
• setAppName (string name) – cài đặt tên ứng dụng cho ứng dụng của bạn.  
• get (string key) - lấy key  
API của StreamingContext:  
Đây là điểm mục chính cho chức năng Spark. Một SparkContext đại diện cho kết nối với một Spark cluster, và có thể được sử dụng để tạo ra các RDD(bộ dữ liệu phân tán có tính đàn hồi), các accumulators và các biến truyền trên cluster. StreamingContext được định nghĩa như sau:

public StreamingContext(String master, String appName, Duration batchDuration,

String sparkHome, scala.collection.Seq<String> jars,

scala.collection.Map<String,String> environment)

• master -cluster URL để kết nối đến (ví dụ mesos: // host: port, spark: // host: port, local [4]).  
• appName - một tên cho công việc của bạn, để hiển thị trên giao diện web cụm  
• batchDuration - khoảng thời gian mà dữ liệu phát trực tuyến sẽ được chia thành các đợt.

public StreamingContext(SparkConf conf, Duration batchDuration)  
  
Tạo một StreamContext bằng cách cung cấp cấu hình cần thiết cho một SparkContext mới.  
• conf - Thông số Spark  
• batchDuration - khoảng thời gian mà dữ liệu phát trực tuyến sẽ được chia thành các đợt  
API KafkaUtils  
API KafkaUtils được sử dụng để kết nối cụm Kafka với Spark streaming. API này có chữ ký createStream signifi -cant được định nghĩa như dưới đây.

public static ReceiverInputDStream<scala.Tuple2<String,String>> createStream(

StreamingContext ssc, String zkQuorum, String groupId,

scala.collection.immutable.Map<String,Object> topics, StorageLevel storageLevel)

Cách trình bày ở trên được sử dụng để Tạo một luồng đầu vào thu hút thông điệp từKafka Brokers.  
• ssc - Đối tượng StreamingContext.  
• zkQuorum - Nhiệm vụ của Zookeeper.  
• groupId - Id nhóm cho người tiêu dùng này.  
• topics - trả lại bản đồ của chủ đề tới consume.  
• storageLevel - Mức lưu trữ để sử dụng để lưu trữ các đối tượng kế thừa.  
API KafkaUtils có một phương thức createDirectStream, được sử dụng để tạo ra một input stream vào trực tiếp thu hút các messages từ Kafka Brokers mà không sử dụng bất kỳ receiver nào. Stream này có thể đảm bảo rằng mỗi messages từ Kafka được bao gồm trong các phép biến đổi chính xác một lần.  
Ứng dụng mẫu được thực hiện trong Scala.

Xây dựng Script

Tích hợp spark-kafka phụ thuộc vào spark, spark streaming and spark Kafka. Các sbt sẽ tải các jar cần thiết trong khi biên dịch và đóng gói các ứng dụng.

name := "Spark Kafka Project"

version := "1.0"

scalaVersion := "2.10.5"

libraryDependencies += "org.apache.spark" %% "spark-core" % "1.6.0"

libraryDependencies += "org.apache.spark" %% "spark-streaming" % "1.6.0"

libraryDependencies += "org.apache.spark" %% "spark-streaming-kafka" % "1.6.0"

Biên soạn / Đóng gói

Chạy lệnh sau để biên dịch và đóng gói tệp tin jar của ứng dụng. Chúng ta cần phải gửi tệp jar vào điều khiển spark để chạy ứng dụng.

sbt package

Gửi đến Spark

Khởi động Kafka Producer CLI , tạo một chủ đề mới được gọi là “my-first-topic” và cung cấp một số thông báo mẫu như dưới đây.

Another spark test message

Chạy lệnh sau để gửi ứng dụng tới spark.

/usr/local/spark/bin/spark-submit --packages org.apache.spark:spark-streaming

-kafka\_2.10:1.6.0 --class "KafkaWordCount" --master local[4] target/scala-2.10/spark

-kafka-project\_2.10-1.0.jar localhost:2181 <group name> <topic name> <number of threads>

Mẫu đầu ra của ứng dụng này được hiển thị bên dưới.

spark console messages ..

(Test,1)

(spark,1)

(another,1)

(message,1)

spark console message ..