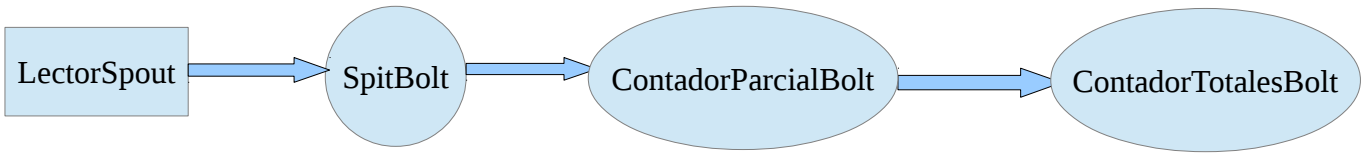
Ejercicio con Storm: Contar Palabras

Para evaluar la parte correspondiente a Storm se propone realizar el siguiente ejercicio. El cual consta de una parte básica y de una serie de variantes propuestas que serán consideradas a la hora de la evaluación.

Desarrolle e implemente una topología Storm formada al menos por un Spout y tres Bolts:

- **LectorSpout**: Se encarga de abrir un fichero de texto (o conectarse a un API para leer secuencias de Strings) y de emitir cada una de las líneas a un Bolt. Puede elegir el fichero que deseé, pero se recomienda que se trate de un fichero con un tamaño considerable. A ser posible que se trate de algún dataset público (por ejemplo los papeles de Panamá).
- **SplitBolt**: Este bolt se encarga de dividir la línea de texto recibida del LectorSpout y emitir palabra por palabra eliminando caracteres de puntuación.
- **ContadorParcialBolt:** Lleva la cuenta del número de palabras totales, el número de palabras únicas y cuales son.
- **ContadorTotalesBolt**: Este bolt realiza la cuenta de las palabras recibidas hasta el momento y las almacena en una BBDD redis, una estructura Hashmap o similiares.
- **ContadorPalabrasTopología** que une los Spouts y los Bolts.

El diagrama de flujo de los Spouts y Bolts sería el siguiente.



Las tuplas de **SplitBolt** a **ContadorParcialBolt** se deben de envíar de forma que las palabras similares le lleguen siempre al mismo Bolt. De está forma cada instancia de ContadorParcialBolt llevará la cuenta de palabras similares. Por ejemplo si SplitBolt emite la siguiente secuencia:

hola que tal estas hola como estas hola, estas

El CondadorParcialBolt deberá obtener los pares:

hola,3
que,1
tal,1
estas,3
como,1

De está forma se obtiene el número de palabrás únicas, sus totales, y el total de palabras (la suma del total de las palabrás únicas). En este ejemplo, habría 5 palabras únicas y 9 palabras totales.

El **ContadorTotalesBolt** recibe todos los contadores Parciales de cada uno de los **ContadorParcialBolts** y por tanto tiene la visión global que debe de almacenar. Este paso se puede ver como un *reduce* en el paradigma map-reduce.

Se pide (parte básica):

1. Utilizando el esquema propuesto, sessarrollar la topología y los Spouts/Bolts.
2. Ejecutar la topología con distinto número de Spouts y Bolts e indicar el tiempo de ejecución a medida que realiza con pruebas con ficheros de mayor tamaño, para medir la escalabilidad de la topología.

Variantes propuestas:

- Utilize una estructura de contador aproximada en los bolts para las palabras.
- Realice un contador no solo de palabras (unigramas) sino también de bi-gramas y tri-gramas.
- Relice la integración de Storm con Kafka y lea las palabras a partir de una cola Kafka.

Envío de ficheros

Por favor envíe los ficheros fuente java necesarios para compilar y ejecutar la topología así como el fichero pom.xml y un documento en formato ascii (txt) con los tiempos de ejecución, todo ello comprimido en un fichero zip con su nombre. Ejemplo: pepe_garcia_storm.zip

La fecha límite de entrega es el Lunes 16 de Mayo a las 23:59.