Pautas generales para las prácticas:

1. Las prácticas **deberán acompañarse de un breve documento README** que detalle cómo ejecutar la práctica y cómo configurar los parámetros si existen, así como cualquier otra particularidad que sea necesario conocer. Adicionalmente, si se desea explicar algún detalle de implementación, podrá adjuntarse opcionalmente un documento de diseño en el que se presente y justifique el diseño elegido.

2. El código debe seguir las **buenas prácticas** habituales: variables con nombres significativos, comentarios, eficiencia, etc.

Ejercicio de evaluación: Hadoop

¡Enhorabuena! Tus aptitudes como programador de Hadoop no han pasado desapercibidas y has ganado un contrato en el equipo de evaluación de Barack Obama. El equipo de imagen ha detectado que las redes sociales y en especial Twitter son un arma indispensable en las campañas electorales modernas.

Te han asignado el proyecto de detectar qué usuarios de Twitter están acaparando la mayor influencia, de cara a tener en cuenta sus comentarios en el próximo discurso del presidente. Para ello, se han diseñado dos medidas de popularidad: Volume Momentum y Popularity Momentum.
Volume Momentum

La fórmula consiste en tomar los últimos N tweets del usuario y hacer el siguiente cálculo:

[(t_N - t_0) - (t_N - t_N/2)]/(t_N - t_0) donde t_N es el timestamp del tweet en la posición N.

Como puedes observar, esta medida captura si la actividad de un usuario se está acelerando siendo próximo a 1.0 cuando existe una aceleración y a 0.0 cuando hay una desaceleración.

Popularity Momentum

La fórmula consiste en tomar los últimos N tweets de un usuario y hacer el siguiente cálculo:

num_retweets_(N/2, N) / totalRetweets_(0,N) donde num_retweets es el número de tweets de un usuario que han sido retwiteados durante el periodo.

Como se puede observar, este índice captura si un usuario esta sufriendo una aceleración en sus últimas aportaciones.

En este caso tomaremos N=20. Las medidas se calculará con los últimos 20 tweets (o los que haya si no llega a 20). Cuando haya un único tweet se tomará 0 como valor para ambas medidas.

El equipo también esta interesado en obtener el número total de tweets de un usuario. Con toda esta información, el equipo estará en disposición de filtrar los influencers a tener en cuenta en el próximo discurso.

Datos disponibles

El equipo quiere utilizar la base de datos disponible en https://datahub.io/dataset/twitter-2012-presidential-election

Requisitos

La salida deseada es un CSV con:

> nombre_usuario, total_tweets, volume_momentum, popularity_momentum

Se deberá entregar un zip con el código fuente en Java de la solución así como responder a la siguiente pregunta:

> ¿Cuáles son las estadísticas de 00ASHLEYMARIE00 y 007_Debby?

Consejos

El ejercicio debería ser un código sencillo si sigues los principios de diseño presentados en los ejercicios. Haz especial hincapié en tener un buen: a) formato de entrada, b) formato de salida, c) ordenación, de manera que el código que tengas que escribir en las fusiones map y reduce sea el más sencillo posible. Recuerda que cada detalle de tu código cuenta, puesto que cada operación se va a ejecutar millones de veces.

Vamos a ceñirnos al fichero cache-1000000-json.gz de 3Gb. Tu código no debería de llevar más de 15 min en ejecutar en un portátil medio moderno. Si lleva más, revisa tu implementación para eliminar los cuellos de botella.

Es posible que necesites incluir alguna librería externa para ayudarte, por ejemplo jackson para parsear entradas JSON. Si estas librerías no están disponibles en el classpath del cluster, es necesario que las incluyas en tu paquete. He aquí un ejemplo de como hacer esto en gradle:

```
configurations {
myDep
}
dependencies {
 myDep group: 'com.fasterxml.jackson.core', name: 'jackson-databind', version: '2.2.4'
 myDep group: 'com.fasterxml.jackson.core', name: 'jackson-core', version: '2.2.4'
 myDep group: 'com.fasterxml.jackson.core', name: 'jackson-annotations', version: '2.2.4'

 compile group: 'com.fasterxml.jackson.core', name: 'jackson-databind', version: '2.2.4'
 compile group: 'com.fasterxml.jackson.core', name: 'jackson-core', version: '2.2.4'
 compile group: 'com.fasterxml.jackson.core', name: 'jackson-annotations', version: '2.2.4'
}
jar {
 from { configurations.myDep.collect { it.isDirectory() ? it : zipTree(it) } }
}
```