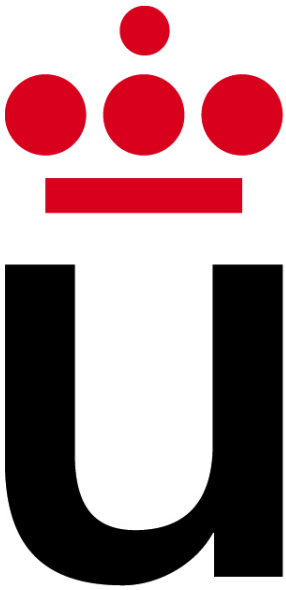
Foro de preguntas

Análisis de Big Data

Preguntas Tema 2

José Ignacio Escribano



Móstoles, 25 de marzo de 2016

Índice

1. Preguntas **1**

1.1. ¿Cuál es el resultado de la ejecución del segundo ejemplo proporcionado en el Tema 2, llamado top10_words.py?¿Qué tareas realiza exactamente este archivo? ¿Por qué la lectura de la salida de un proceso con mrjob se debe poner en un archivo diferente (top10_words.py) de aquel en que declaramos la clase principal de la tarea (MRCountWords.py)? 1

1.2. Las dos fases principales en cualquier tarea MapReduce son map y reduce. Sin embargo, también se puede añadir un nuevo tipo de fase llamada combine. Busque información sobre este tipo de fase y explique con un ejemplo qué ventajas proporciona cuando se añade a ciertos procesos MapReduce. 2

1. Preguntas

1.1. ¿Cuál es el resultado de la ejecución del segundo ejemplo proporcionado en el Tema 2, llamado top10_words.py?¿Qué tareas realiza exactamente este archivo? ¿Por qué la lectura de la salida de un proceso con mrjob se debe poner en un archivo diferente (top10_words.py) de aquel en que declaramos la clase principal de la tarea (MRCountWords.py)?

La salida del script "top10_words.py" pasarle como parámetro el fichero "shakespeare.txt" es la siguiente:

```
[[27801, 'the'], [26834, 'and'], [20296, 'i'], [19748, 'to'],
[18299, 'of'], [14620, 'a'], [13713, 'you'], [12474, 'my'],
[11149, 'that'], [11060, 'in']]
```

Es decir, este script muestra las diez palabras más frecuentes del archivo que se le pasa como parámetro, junto con el número de apariciones de cada palabra.

En el archivo "shakespeare.txt" la palabra más frecuente es "the" con 27801 apariciones en el texto. "And" es la segunda palabra más frecuente con 26834 apariciones. En décimo lugar, se encuentra la palabra "in" con 11060 apariciones.

Este archivo instancia un objeto de la clase MRCountWords, que es a su vez un mrjob. Llegan tuplas de la forma (None, [palabra, frecuencia]) procedentes de la función reduce. Por cada una de estas tuplas, se selecciona el segundo miembro, que contiene la palabra y su frecuencia. Cada una de estos vectores es guardado en una lista. Una vez ha terminado el proceso, se ordena la lista de forma descendente y se seleccionan los diez primeros.

No se puede tener la lectura de la salida de un proceso mrjob y la clase principal de la tarea, ya que el archivo con la clase tarea se envía a Hadoop para que corra. Por lo tanto, el archivo con la tarea no puede intentar iniciar el job de Hadoop, ya que estaría creando jobs de Hadoop de forma recursiva[1].

Fuente: [1]: http://pythonhosted.org/mrjob/guides/runners.html#why-not-runner-in-file

1.2. Las dos fases principales en cualquier tarea MapReduce son map y reduce. Sin embargo, también se puede añadir un nuevo tipo de fase llamada combine. Busque información sobre este tipo de fase y explique con un ejemplo qué ventajas proporciona cuando se añade a ciertos procesos MapReduce.

La fase combine se usa entre la clase map y reduce para minimizar el volumen de datos (pares clave-valor) transferidos entre estas dos fases.

Esta fase es adecuada cuando se quiera aplicar a una función que sea conmutativa y asociativa a la vez.

Consideremos el ejemplo de contar palabras de un texto. Si no usáramos la fase combine, al final de la etapa map tendríamos tuplas de la forma
(palabra 1, 1)
(palabra 1, 1)
(palabra 2, 1)
...

Si tuviéramos un archivo muy grande, las tuplas con las palabras más frecuentes se enviarían a la fase reduce muchísimas veces.

Si usamos la fase combine, tendríamos tuplas de la forma
(palabra 1, N)
(palabra 2, M)
...

donde N, M son números mucho más grandes que 1.

De forma, es posible reducir, o incluso minimizar el intercambio de información entre las fases map y reduce.

Fuentes:

[1]: http://www.tutorialspoint.com/map_reduce/map_reduce_combiners.htm
[2]: http://www.philippeadjiman.com/blog/2010/01/14/hadoop-tutorial-series-