Producto Cartesiano

En teoría de conjuntos, el producto cartesiano de dos conjuntos es el conjunto de todos los pares ordenados que pueden formarse tomando el primer elemento del primer conjunto, y el segundo elemento del segundo conjunto.

En el contexto computacional, una operación de producto cartesiano suele ser extremadamente costosa en términos de tiempo de ejecución y lectura de disco. Las ventajas que map/reduce ofrece para otras construcciones (split de input, paralelización) no pueden aplicarse en este caso.

Descripción del problema y datos de entrada

Dado un archivo de comentarios en formato xml, el programa deberá seleccionar los pares que tengan al menos cierto número de palabras en común.

Como datos de entrada, se usó uno de los archivos del challenge anterior (comentarios en sitios de StackExchange, extraídos del *Stack Exchange Data Dump March 2013*). En este caso se usó el archivo de meta.scifi.stackexchange.com (1.8 MB – 1850 usuarios – 6,227 comentarios).

Descripción de la implementación

La clase CartesianInputFormat se encarga de la producción de los splits para la entrada. Cada InputSplit retornado es un par de splits, cada uno generado por medio de un FileInputSplit estándar a partir de la lectura de ambos archivos (el mismo en este caso).

Gran parte de la lógica está en la clase CartesianRecordReader. Este reader recibe cada CompositeInputSplit generado por el InputFormat y obtiene un RecordReader (secuencial) a partir del primer FileInputFormat del composite. Luego, *para cada ítem* retornado por este primer Reader, construye un segundo RecordReader para el segundo archivo, e *itera completamente por el mismo*. De esta forma se generan pares clave-valor en que la clave son todos los registros del primer archivo, y los valores todos los registros del segundo.

El mapper en sí es muy sencillo, simplemente toma la clave y valor de los pares de entrada (dos comentarios) y convierte cada uno a un conjunto de sus palabras. Luego verifica si la cantidad de

ítems en común entre ambos conjuntos es mayor al límite establecido. En caso que esto ocurra se emite la lista de palabras en común y el texto completo de ambos comentarios.[1]

Similitud con el self-join

El producto cartesiano es esencialmente igual al self-join, salvo que por lo general el término self-join se utiliza cuando existe una clave común para la elección de los pares (por ejemplo: todos los pares de comentarios *del mismo usuario*). El producto cartesiano, en cambio, produce todos los pares posibles.

Resultados

Como era de esperarse, la ejecución del producto cartesiano resulta ser extremadamente costosa, aún en el caso de un archivo de entrada pequeño. El tiempo insumido fue de casi **19 minutos** (aún con la modificación para que el archivo de salida no fuese tan grande) y se leyeron de disco 11695961218 bytes (lo cual corresponde perfectamente a procesar 6227 veces un archivo de 1.8 MB).

[1] Sólo a los efectos de no generar un archive de salida tan grande, se modificó este código para incluir en el output únicamente las palabras en común y el id de ambos comentarios.