**MapReduce, algunos apuntes**

Muchos problemas pueden resolverse mediante el modelo de map-reduce, pero también hay muchos que no. Por eso conviene centrarse un poco en las ideas conceptuales del esquema de programación de MapReduce.

Este modelo procede de la programación funcional y consiste en expresar un problema en dos operaciones:

* Map, que aplica una misma función a una lista de elementos
* Reduce, que combina la lista de los resultados producidos por map.

Veamos dos ejemplos sencillos con los que introducir los conceptos básicos uno a uno:

* El del sumatorio

Una observación crucial es que la función map se puede aplicar a los elementos de la lista en cualquier orden, y por tanto, también en paralelo. A la función que se aplica a los elementos se le suele llamar mapper, función mapeadora. En nuestro caso, la función mapeadora calcula el elemento k-ésimo de la secuencia, para cada k.

La función reductora, el reducer, ha de ser conmutativa y asociativa. De este modo, los pares de resultados que da la función map se pueden combinar en cualquier orden, y en particular en paralelo y a medida que vayan siendo generados por los mappers

* El de los párrafos

Conviene observar la independencia del orden de procesamiento de cada párrafo, y de la posterior combinación de los resultados.

En realidad, la función map en el modelo MapReduce hace algo más que procesar elemento a elemento:

Map toma cada elemento de la lista de entrada y devuelve una lista de pares, clave-valor en un dominio diferente:

Map(f,[alpha]) -> [(clave, beta)]

…………………………………………………………………………………………………………………………..

El modelo de mapreduce está pensado para que una tarea se procese en un clúster formado por distintas máquinas. Desde el punto de vista práctico, la cosa funciona como sigue:

* Los datos de entrada proceden por regla general de un archivo, normalmente de gran tamaño, y se dividen en un conjunto de *particiones* de entrada, generalmente de entre 16 y 64 Mb, y cada una de estas particiones se va enviando a una máquina del clúster que esté en reposo.
* Cada máquina recibe su bloque de información y los procesa independiente de las demás, aplicando el mapper.
* Cada resultado tiene la forma clave-valor, como se ha explicado. Los pares clave y valor producidos por las distintas máquinas se van agrupando en regiones y enviados nuevamente a máquinas con la finalidad de aplicarles las tereas de reductoras.
* Las claves juegan un papel esencial, porque frecuentemente muchas claves permiten agrupar los resultados de los mappers para su procesamiento; el ejemplo segundo (pares de nombres propios) evidencia este hecho.