Pràctica 1: Models de comunicació y Middleware

Alumnes: Tomàs Biarnés Gaig

David Sánchez Benaiges

Assignatura: Sistemes Distribuïts

Curs: 2018-19

Índex

1. Introducció

MapReduce és un model de programació i implementació que permet el processament paral·lel de grans quantitats de dades. En poques paraules, trenca un gran conjunt de dades en trossos més petits per processar-los per separat en diferents nodes i recopila automàticament els resultats per retornar un sol resultat.

Com el seu nom indica, permet processar el mapa i reduir les operacions funcionals, que realitzen la major part de la lògica de programació. De fet, consta de dos grans passos, que són els següents:

Pas "Map": cada node aplica la funció "map ()" a les dades locals i escriu la sortida a un emmagatzematge temporal. Un node mestre assegura que només es processa una còpia de les dades d'entrada.

Pas "Reduce": els nodes ara processen cada grup de dades de sortida, per clau, i el nombre d'aparicions, per valor, en paral·lel.

Atès que cada operació de mapatge és independent de la resta, tots els mapes es poden realitzar de forma paral·lela. El mateix passa amb els reductors, ja que totes les sortides de l'operació del mapa es lliuren al mateix reductor.

2. Implementació realitzada

L' implementació consisteix en la creació de un nombre determinat de mappers, els quals realitzaran la tasca de llegir una part del fitxer, creant un diccionari amb les paraules que s' han trobat en la partició corresponent, aquest es puja a COS.

Una vegada el reducer ha trobat els diccionaris al COS, s' uneixen i es forma un del sol. En aquest mateix pas es conten les paraules totals del diccionari per tenir el CountWords. Una vegada acabada l' acció, aquesta penja el diccionari de les paraules trobades a COS.

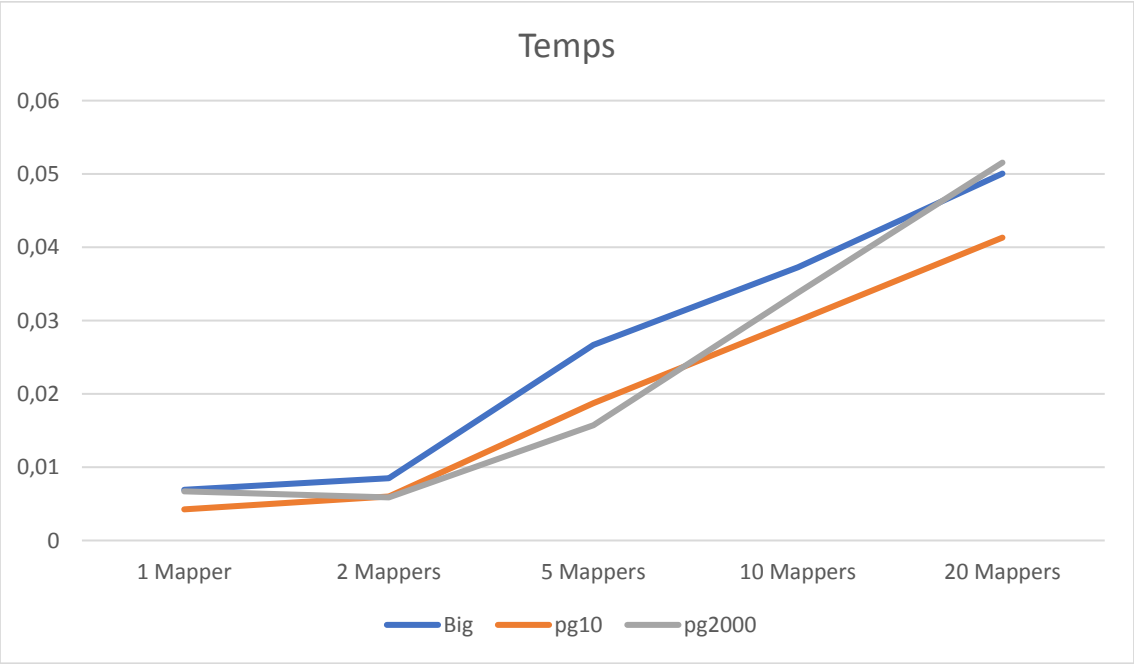
S'ha de tenir en compte que per poder llançar el nostre mapreduce de manera satisfactòria s'ha de realitzar una modificació al fitxer ibm_cloud_config, afegint un paràmetre mes al ibm_cos, el paràmetre s'ha de dir "repositori" i el valor ha de ser el nom del bucket del COS.

Per llançar el programa hem d'indicar els paràmetres de nom de fitxer i nombre de mappers, quedant de la següent manera: "python orchestrator.py nom.txt 5".

3. Joc de proves

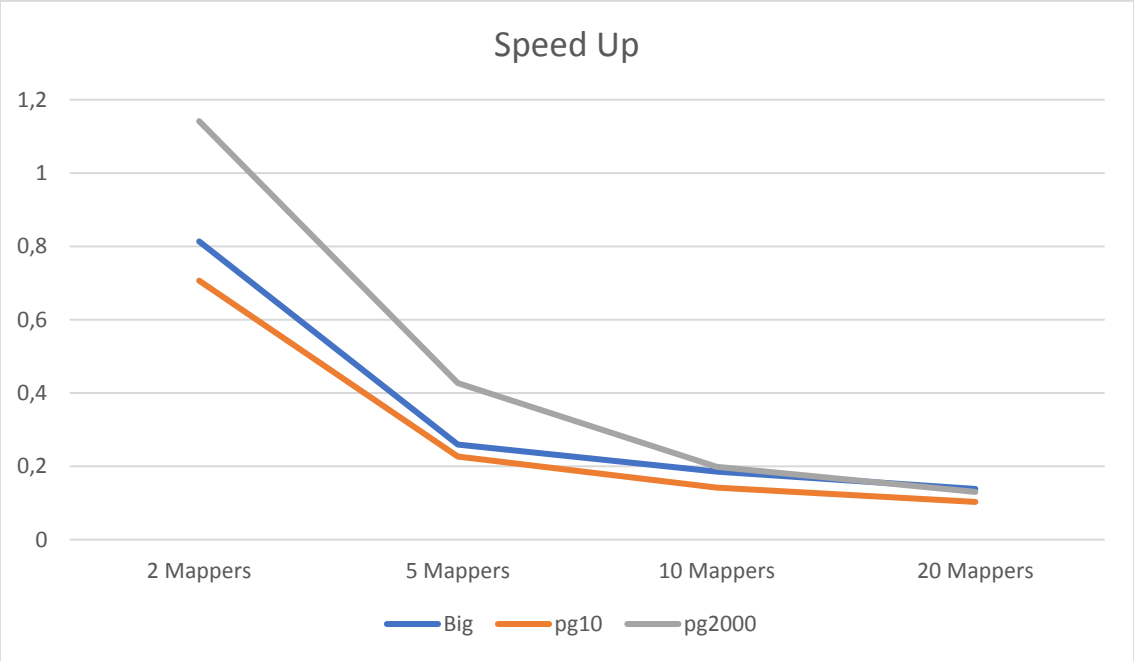
Fitxer	Implementació	Temps
Big.txt	1 mapper	0.006917
Pg10.txt	1 mapper	0.004247
Pg2000.txt	1 mapper	0.006709
Big.txt	2 mappers	0.008503
Pg10.txt	2 mappers	0.006012
Pg2000.txt	2 mappers	0.005877
Big.txt	5 mappers	0.026685
Pg10.txt	5 mappers	0.018734
Pg2000.txt	5 mappers	0.015709
Big.txt	10 mappers	0.037263
Pg10.txt	10 mappers	0.029969
Pg2000.txt	10 mappers	0.033798
Big.txt	20 mappers	0.050047
Pg10.txt	20 mappers	0.041311
Pg2000.txt	20 mappers	0.051548

A continuació s'observa una gràfica del temps d'execució de cada un dels fitxers segons el nombre de mappers que l'han analitzat.



Veiem que el temps augmenta linealment a partir de l' utilització de dos mappers,

Tot seguit veiem el gràfic amb el càlcul de l'evolució del Speed-up segons el nombre de mappers per cada fitxer.



Podem observar que el nostre codi només té un Speed-up positiu en el cas de 2 mappers pel fitxer pg2000 en la resta de fitxers i nombre de mappers perd rendiment.