

Laboratori 5

Daniel Beltrán Drago y Toni Cifré Vicens

16 Desembre 2020

Contents

1	Explicació de les classes	3
2	Preguntes	4
3	Anàlisi d'execució en funció del nombre de nuclis.	5

1 Explicació de les classes

MRMarketBasket1

Aquesta classe està dissenyada per a cridar de forma ordenada al **mapper**, **combiner** i al **reducer** amb l'objectiu d'obtenir com a resultat el nombre d'associacions que es produeixen a l'hora de comprar un conjunt de productes. La funció del **mapper** divideix cada línia utilitzant com a delimitador la coma d'aquesta, un cop separat ordena les paraules per ordre alfabètic per tal que les claus sempre apareguin en el mateix ordre, a continuació genera les permutacions de totes les parelles de productes possibles i les envia cap al **combiner**. En aquest cas el **combiner** fa la mateixa tasca que el **reducer**, es a dir, per cada una de les claus (conjunt de dos productes) suma la llista de valors donats per paràmetre això significa que retornarà la clau i el nombre de cops que apareix aquesta relació.

En aquesta classe el **combiner** no generarà una gran reducció en el cost de l'execució perquè no hi ha un gran nombre de parelles generades per cada un dels threads.

MRMarketBasket2

En aquest cas la classe **MRMarketBasket2** s'encarrega de sumar tots els cops que es compren cada un dels productes. Igual que en la classe anterior, també és cridar de forma ordenada al **mapper**, **combiner** i al **reducer**. El **mapper** un cop ha dividit cada un dels productes, retorna el nom com a clau i un 1 com a valor, el **combiner** suma quants de cops s'ha trobat cada un dels productes en el mateix thread de la crida i finalment el **reducer** retorna la suma total per a cada un dels productes.

MRMarketBasketAnalysis

Per dur a terme l'anàlisi s'ha utilitzat la llibreria **Pandas** per a guardar el resultat obtingut per la criada a **MRMarketBasket1** i en el cas de **MRMarketBasket2** s'ha utilitzat un diccionari. Un cop s'ha obtingut els resultats de les crides i amb la utilització de la llibreria **Pandas** hem dividit el parell de claus retornat per la primera crida i hem calculat el suport i cada un dels valors de confiança per al parell de productes. Un cop calculat els valors de la confiança i el suport hem comptat quantes regles d'associació estan dintre dels paràmetres de confiança i suport donats.

2 Preguntes

A partir de la vostra implementació, ompliu la taula següent; els valors de la primera fila ja se us donen calculats

<i>Fila</i>	<i>Support</i>	<i>Confiança</i>	<i>Nre. de regles d'associació trobades</i>
1	1%	1%	426
2	1%	25%	96
3	1%	50%	0
4	1%	75%	0
5	5%	25%	4
6	7%	25%	2
7	20%	25%	0
8	50%	25%	0

Doneu la llista de regles d'associació que hàgiu trobat corresponents a les files 4, 5 i 6 de la taula de la pregunta 1

Fila 4

No te taula donat que el nombre de regles d'associació trobades és 0

Fila 5

<i>Product 1</i>	<i>Product 2</i>	<i>Support</i>	<i>Confidence</i>
whole milk	other vegetables	0.07483477376715811	0.29287703939514526
other vegetables	whole milk	0.07483477376715811	0.38675775091960063
rolls/buns	whole milk	0.05663446873411286	0.30790491984521834
yogurt	whole milk	0.05602440264361973	0.40160349854227406

Fila 6

<i>Product 1</i>	<i>Product 2</i>	<i>Support</i>	<i>Confidence</i>
whole milk	other vegetables	0.07483477376715811	0.29287703939514526
other vegetables	whole milk	0.07483477376715811	0.38675775091960063

3 Anàlisis d'execució en funció del nombre de nuclis.

Com es pot observar en la taula, el que nosaltres busquem en paral·lelitzar l'execució és obtenir una disminució del temps d'execució, però en aquest cas he hagut d'executar el programa en una màquina virtual de Linux perquè en Windows el funcionament no era correcte. Aquesta màquina virtual només disposava de 2 nuclis, és a dir que com més nuclis per sobre dels que la màquina disposa produeix un coll de botella en l'execució, ja que es divideix en nuclis no existents i han d'esperar que els nuclis disponibles acabin primer el que tenen assignat.

<i>N Cores</i>	<i>Temps</i>
1	8.4422
2	8.0417
3	9.1236
4	11.4497
5	14.4479

En cas d'executar el codi en una màquina amb més cores, podríem observar com amb més cores executem el programa, menys triga respecte a una execució lineal, però nosaltres no hem pogut observar-ho de forma directa.