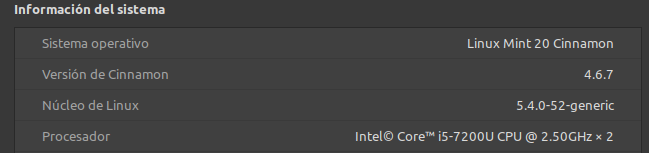
# Pràctica 6: Map Reduce and Document Clustering

1. Introducció

Per a la realització de la pràctica utilitzem el mateix dispositiu per a que no es vegin afectats els resultats obtinguts. L’ordinador amb el qual hem treballat consta de les característiques següents:



Com es pot veure a les espicificacions anteriors aquest només consta de 2 cores: això afectarà a l'elecció dels fitxers a experimentar i als experiments relacionats amb els ncores empleats per a l’execució. Així, per adaptar-nos de la millor manera hem treballat amb els fitxers de la carpeta 20\_newsgroups i hem comparat les execucions en el cas d’usar només 1 core o en canvi usar-ne 2.

A més a més, hem considerat oportú canviar el nombre d’iteracions a 50 per a assegurar-nos que l’execució acabarà quan les dues execucions finals convergeixin.

1. Experimentació i conclusions

Per a tots els experiments hem usat nclust = 8 en la generació del nombre de clusters inicials. A mesura que hem avançat els experiments, ens hem anat quedant amb els paràmetres que semblaven donar els resultats més coherents.

* 1. Experimentant amb les freqüències mínimes i màximes

| Freqüències  (min - max) | 0.025 - 0.075 | 0.03 - 0.08 | 0.05 - 0.15 | 0.1 - 0.3 | 0.2 - 0.6 |
| --- | --- | --- | --- | --- | --- |
| Número d’iteracions | 2 | 2 | 4 | 4 | 4 |
| Temps primera iteració (segons) | 2.00 | 1.90 | 2.26 | 3.07 | 2.52 |
| Temps última iteració (segons) | 5.63 | 5.29 | 2.80 | 2.14 | 1.88 |
| Temps màxim (iteració : segons) | 2 : 5.63 | 2 : 5.29 | 2 : 7.46 | 2 : 7.61 | 2 : 4.22 |
| Número final de clusters | 8 | 8 | 2 | 1 | 1 |

Per estudiar l’efecte de la freqüència mínima i màxima de les paraules sobre els resultats, hem provat diverses freqüències, començant per les inicials que ens proporciona l’enunciat: 0.1 de freq mínima, 0.3 de freq màxima. Per aquests valors podem observar que l’agrupació final resulta en 2 clusters. En aquest punt hem pensat que aquesta agrupació pot donar-se pel fet que les freqüències són massa altes, per tant 1 sol prototip és suficient per representar tots els documents. Per comprovar que, efectivament, a partir d’aquí les freqüències majors no divideixen els documents en més agrupacions, hem provat freqüències més altes, i el resultat és l’esperat (1 cluster).

A partir d’aquí ens hem centrat en explorar freqüències més petites, cercant un punt intermig on el nombre de clusters sigues un número entre 1 i 8 (no inclosos). Provant la meitat de les freqüències inicials, 0.05 i 0.15 com a freqüència mínima i màxima respectivament, trobem uns resultats diferents: 2 agrupacions finals. Això significa que les paraules que s’escullen per representar els prototips són més diferencials que les anteriors. A continuació, hem volgut experimentar amb freqüències encara menors per veure si les agrupacions es troben afectades. Primer hem testejat, la meitat de les freqüències anteriors, però el número de clusters final ha sigut el mateix que l’inicial, 8. En segon lloc, hem experimentat amb unes freqüències intermitges entre les dues proves anteriors, però no hem aconseguit reduir el nombre d’agrupacions final.

Per tant, concluïm que les millors freqüències són 0.05 de mínima i 0.15 de màxima, sent les paraules amb freqüències menors massa diferencials per agrupar correctament els documents. Per les pròximes experimentacions ens quedarem amb aquests paràmetres.

Per últim, si ens fixem en els temps d’execució i el nombre d’iteracions, observem que per a freqüències majors el nombre d’iteracions augmenta. Els temps de les iteracions inicials són baixos en tots els casos (es tracta només un document). El temps màxim s’aconsegueix en la segona iteració per a totes les proves, al passar a tractar tots els documents, i és major per a freqüències més elevades (exceptuant l’última prova). Els temps finals també són baixos, que és on es troba la convergència entre iteracions.

* 1. Experimentant amb número de paraules

| Número de paraules | 100 | 175 | 250 |
| --- | --- | --- | --- |
| Número d’iteracions | 4 | 4 | 4 |
| Temps primera iteració (segons) | 2.12 | 2.28 | 2.55 |
| Temps última iteració (segons) | 2.16 | 2.73 | 2.14 |
| Temps màxim (iteració : segons) | 2 : 3.67 | 2 : 5.55 | 2 : 8.55 |
| Número final de clusters | 2 | 2 | 2 |

Per comprovar que el cost computacional empitjora en els casos on tractem amb més paraules, hem provat d’estudiar els casos amb 100, 175 i 250 paraules al seu vocabulari. Com podem veure a la taula de resultats aquest fenòmen es veu reflectit a les iteracions intermitges, no a les dels extrems. La iteració amb temps màxim mostra el creixement del cost computacional que suposa augmentar el nombre de paraules que es tracten, arribant a 8.55 segons per a 250 paraules. També observem que aquest augment no es veu reflectit en la primera ni la última iteració de cada execució, degut a que al principi només es tracta un document per a totes les proves.

* 1. Experimentant amb número ncores (número de mappers i reducers)

| Ncores | 1 | | 2 | |
| --- | --- | --- | --- | --- |
| # Paraules vocabulari | 100 | 250 | 100 | 250 |
| Número d’iteracions | 4 | 5 | 4 | 4 |
| Temps primera iteració (segons) | 2.48 | 3.28 | 2.12 | 2.55 |
| Temps última iteració (segons) | 2.92 | 4.34 | 2.16 | 2.14 |
| Temps màxim (iteració : segons) | 2 : 7.27 | 2 : 15.98 | 2 : 3.67 | 2 : 8.55 |
| Número final de clusters | 2 | 1 | 2 | 2 |

Com hem mencionat a la introducció només disposem de 2 cores i per tant només podem fer la comparació de l’execució en secuencial vs la paral·lela amb 2 cores. Tot i així, hem vist una gran diferència entre ambdues. Com es pot veure a la taula d’adalt si comparem el cost computacional entre els roses i els blaus de les diferents execucions trobem una diferència de fins a 7,4 segons per als temps màxims. Els temps dels extrems es mantenen semblants, com hem mencionat anteriorment. La resta de dades no presenta canvis significatius.

* 1. Experimentant final - Anàlisis de les paraules més freqüents

Per aquest últim experiment hem estudiat la qualitat de les agrupacions finals dels documents, observant les paraules més freqüents dels prototips resultants. En primer lloc, hem analitzat el resultat amb les freqüències òptimes de 0.05 i 0.15 i fixant-nos en 200 paraules més freqüents. Com a cada execució el resultat de K-means varia, hem optat per realitzar més d’una execució i comprovar que la representació de cada cluster és coherent.

FREQÜÈNCIA 0.05-0.15, 200 PARAULES

CLASS2

[(1.0, 'price'), (0.21235521235521235, 'buy'), (0.19594594594594594, 'comput'), (0.18532818532818532, 'list'), (0.1833976833976834, 'drive')]

CLASS3

[(1.0, 'appreci'), (0.166988416988417, 'window'), (0.1303088803088803, 'comput'), (0.12934362934362933, 'email'), (0.12258687258687259, 'advanc')]

CLASS7

[(0.7439261418853256, 'god'), (0.5301263362487852, 'christian'), (0.25072886297376096, 'exist'), (0.24635568513119532, 'him'), (0.24538386783284744, 'word')]

CLASS1

[(0.08755789607258423, 'program'), (0.08755789607258423, 'happen'), (0.08736755282025252, 'comput'), (0.08489309053994036, 'window'), (0.08470274728760865, 'line')]

**prototype-final → 4 clusters**

CLASS3

[(1.0, 'issu'), (0.26375282592313487, 'discuss'), (0.25772418990203466, 'govern'), (0.2569706103993971, 'group'), (0.2554634513941221, 'support')]

CLASS5

[(0.08726567550096961, 'comput'), (0.08581124757595346, 'didn'), (0.08516483516483517, 'window'), (0.08505709976298212, 'opinion'), (0.0850032320620556, 'happen')]

**prototype-final → 2 clusters**

CLASS2

[(1.0, 'game'), (0.3861090365944735, 'play'), (0.21956684092606424, 'him'), (0.17251680358476476, 'best'), (0.1717699775952203, 'didn')]

CLASS5

[(1.0, 'bad'), (0.21019108280254778, 'happen'), (0.2054140127388535, 'again'), (0.20302547770700638, 'enough'), (0.19187898089171976, 'live')]

CLASS1

[(0.08967391304347826, 'support'), (0.0889222941720629, 'program'), (0.088286308973173, 'comput'), (0.08788159111933395, 'opinion'), (0.08765032377428307, 'window')]

**prototype-final → 3 clusters**

Podem notar que en les 3 execucions s’obtenen grups amb paraules bastant relacionades. En la primera execució, per exemple, tenim 2 grups amb paraules significatives, la CLASS 7 conté documents que parlen sobre religió (paraules com “christian” i “god”) i la CLASS 2 agrupa documents sobre compres, amb paraules com “price”, “buy” i “list”.

La següent execució només obté 2 grups, però podem veure que també mostra temàtiques diferents: un sobre política (paraules com “govern”, “discuss”) i l’altre sobre tecnologia (paraules com “comput” i “window”).

La última iteració reflexa també la validesa dels agrupaments amb aquestes freqüències. Un exemple és el cluster 2, del qual podem deduir que parla sobre videojocs, ja que conté paraules com “game” i “play”.

Per últim hem volgut comprovar que, efectivament, si les paraules seleccionades són molt freqüents, tots els documents són representats amb les mateixes paraules i per tant en un sol cluster. En canvi, si usem paraules molt poc freqüents tots els documents tindran paraules diferents. Els clusters resultants en ambdós casos són, en general, random.

Primerament comprovarem el cas en el qual la freqüència de les paraules és molt baixa:

FREQÜÈNCIA 0.025-0.075 200 WORDS

CLASS4

[(0.11291550367926922, 'applic'), (0.11139304744988582, 'copi'), (0.10657193605683836, 'technolog'), (0.10504947982745497, 'org'), (0.09363105810707942, 'board')]

CLASS5

[(0.38134592680047225, 'repli'), (0.1959858323494687, 'chip'), (0.1782762691853601, 'model'), (0.14757969303423848, 'pc'), (0.12278630460448642, 'newsgroup')]

CLASS6

[(0.08242258652094718, 'win'), (0.08151183970856102, 'hold'), (0.08136004857316333, 'anybodi'), (0.07893139040680024, 'her'), (0.07832422586520947, 'although')]

CLASS0

[(0.4318181818181818, 'michael'), (0.4150717703349282, 'disclaim'), (0.2284688995215311, 'stand'), (0.1076555023923445, 'mine'), (0.08253588516746412, 'later')]

CLASS1

[(0.07597402597402597, 'pass'), (0.07077922077922078, 'polit'), (0.06883116883116883, 'offic'), (0.06753246753246753, 'paul'), (0.06753246753246753, 'histori')]

CLASS2

[(0.35478260869565215, 'gun'), (0.2843478260869565, 'dead'), (0.20695652173913043, 'statement'), (0.19130434782608696, 'goe'), (0.18869565217391304, 'doubt')]

CLASS3

[(0.10895289127037285, 'scienc'), (0.082161196695691, 'info'), (0.0808216119669569, 'basic'), (0.08037508372404555, 'present'), (0.07970529135967849, 'receiv')]

**prototype-final → 8 clusters**

En aquest cas les paraules a priori dins la mateixa classe no tenen molta relació, però trobem algunes excepcions en alguns grups com el 5 que conté les paraules “chip” i “pc”. No obstant, costa de deduir la temàtica de cada cluster degut a la mala agrupació dels documents.

Per a finalitzar tractem les paraules amb freqüència molt alta:

FREQÜÈNCIA 0.2-0.6 200 WORDS

CLASS2

[(0.36139948093431823, 'ani'), (0.34762427630265524, 'has'), (0.3475743661409463, 'me'), (0.34642643242164106, 'use'), (0.3450788580555001, 'like')]

**prototype-final → 1 cluster**

Ara trobem només una classe amb les paraules més freqüents. Com era d’esperar aquestes no tenen cap relació entre elles ja que com es pot veure són paraules que usem molt sovint quan redactem o parlem: verbs comuns (“has”, “use”, “like”) i pronoms (“me”).

1. Dificultats que hem trobat

En aquesta pràctica la major dificultat ha estat a la part de la implementació. Comprendre el funcionament del MapReduce i la relació entre els codis MRKmeans i MRKmeansSteps, com la comunicació entre codis mitjançant els documents, han estat les tasques que ens han suposat més temps i esforç.

Després, al moment de l'anàlisi ens hem trobat amb la dificultat de comparar els resultats de les execucions amb diferents ncores ja que el dispositiu amb el qual treballem només disposa de 2, per tant només hem pogut comparar el cas en el qual s’executa en paral·lel en ambdós cores i el cas en el qual es fa de forma seqüencial.