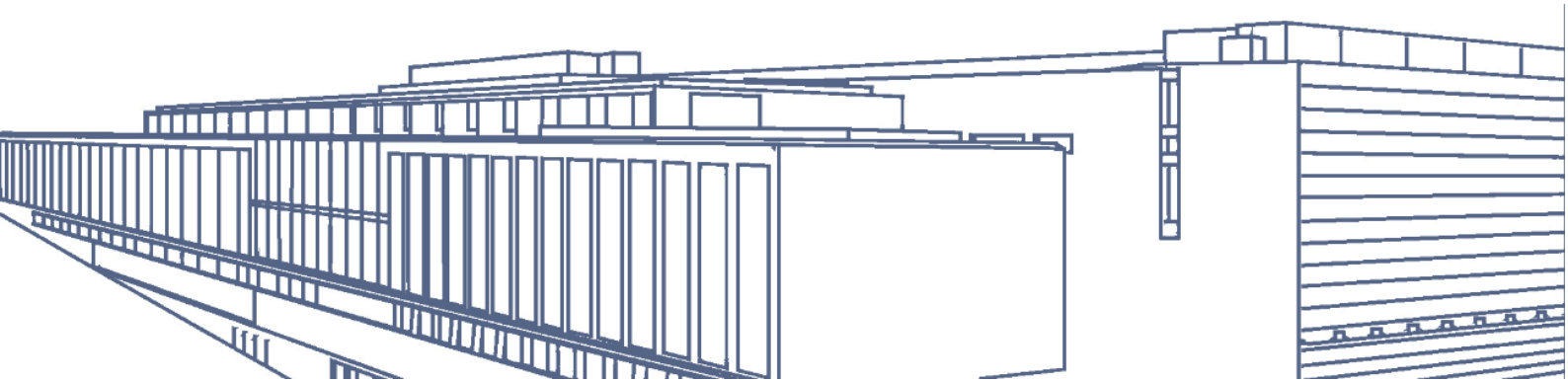
ESCOLA TÈCNICA SUPERIOR
D'ENGINYERIA
Universitat Rovira i Virgili

Sistemes distribuïts

Task 1: Communication models and Middleware

23/04/2019

Victor López Romero
Marcel Magrané Aranda



Contingut

1. Directori GitHub

Aquest treball es pot trobar al següent enllaç de GitHub:
https://github.com/MarcelMagrane/Prac1SD

2. Definició fitxers

2.1. Fitxers de text:

Aquests fitxers contenent cadascun un text diferent, utilitzat per poder realitzar proves.

A continuació s'indiquen els noms i mides de cada fitxer:

Bible.txt -> 4.15 MB.

Quijote.txt -> 2.06 MB.

Sherlock.txt -> 6.19 MB.

2.2. Fitxers amb accions:

Aquests ZIP contenen cadascun els fitxers per tal de realitzar diferents accions. Tots estan formats per tres fitxers:

__main__.py -> Fitxer amb el *main* de l'acció.

cos_backend.py -> Fitxer per tal d'utilitzar el *bucket*.

(*nom acció*).py -> Fitxer amb el codi de l'acció.

Accions que realitzen:

countwords.zip -> Compte el nombre total de paraules en un fitxer de text.

wordcount.zip -> Compte el nombre total de vegades d'aparicions de cada paraula en un fitxer de text.

reduce.zip -> Agrupa els resultats obtinguts als diferents Word Counts utilitzats.

2.3. Fitxers complementaris:

cos_backend.py -> Conté les funcions per tal de pujar, obtenir, obtenir la capçalera i eliminar objectes del *bucket*. També conté la funció per obtenir una llista dels objectes continguts al *bucket*.

ibm_cf_connector.py -> Conté les funcions per tal de crear, obtenir, eliminar i invocar les accions creades al *cloud*.

ibm_cloud_config.py -> Conté la informació personal del cloud (no indicada per privacitat).

2.4. Fitxer principal:

orquestator.py -> Fitxer principal del programa. Crea les accions al *cloud,* i utilitza aquestes accions per executar el programa.

3. Aclariments sobre el disseny

En aquest apartat s'aclariran algunes parts del codi que poden no estar clares o que s'han afegit per criteri dels programadors.

3.1. Orquestator

3.1.1. Realitzar esperes al codi (*waits*)

Al realitzar algunes accions del *cloud* es realitzen uns *waits*. Això s'ha programat d'aquesta manera ja que pot donar-se el cas de que es cridin altres accions sense acabar les anterior que són necessàries per aquesta última.

Waits realitzats:

```
1.  #Wait for wordcount
2.  print("Waiting for the {0} wordcounts to finish:".format(partitions))
3.  end = len(cos.list_objects('sdprac1', 'wc'))
4.  while end != partitions:
5.      end = len(cos.list_objects('sdprac1', 'wc'))
```

En aquest esperem a que els Word Count acabin. Això ho comprovem comparant el nombre de 'wc' (fitxer que crea el Word Count quan acaba) *al cloud* amb el nombre de Word Count que s'han iniciat. Fins que aquest dos valors no siguin iguals el programa no continua.

```
1.  #Wait for reduce
2.  print("Waiting for reduce to finish")
3.  end = len(cos.list_objects('sdprac1', 'reduce'))
4.  while end != 1:
5.      end = len(cos.list_objects('sdprac1', 'reduce'))
```

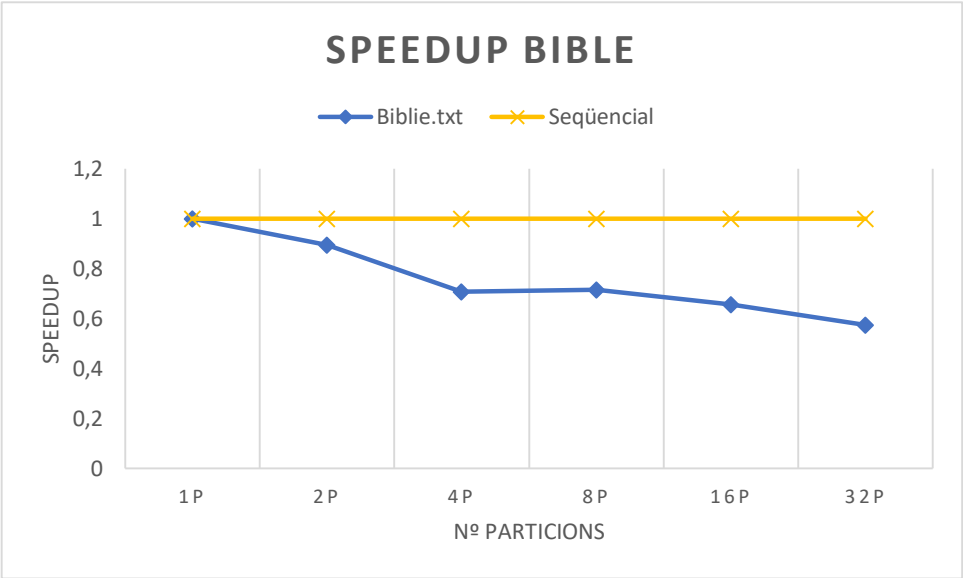
En aquest esperem a que el Reduce acabi. En aquest cas només hi ha un Reduce per tant esperem a que es creí el 'reduce' (fitxer que crea el Reduce quan acaba) al *cloud.*

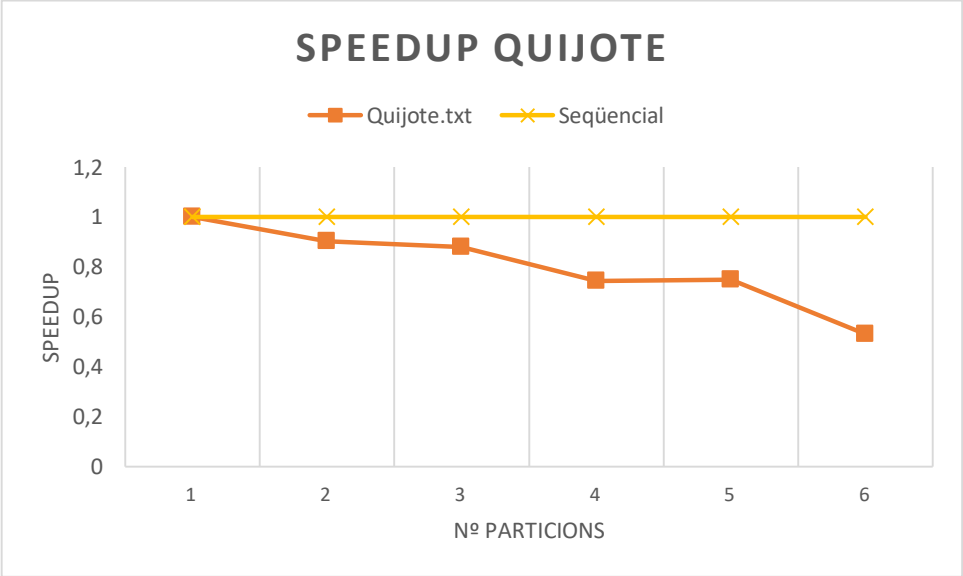
```
1.  #Wait for countwords
2.  print("Waiting for countwords to finish")
3.  end = len(cos.list_objects('sdprac1', 'count'))
4.  while end != 1:
5.      end = len(cos.list_objects('sdprac1', 'count'))
```

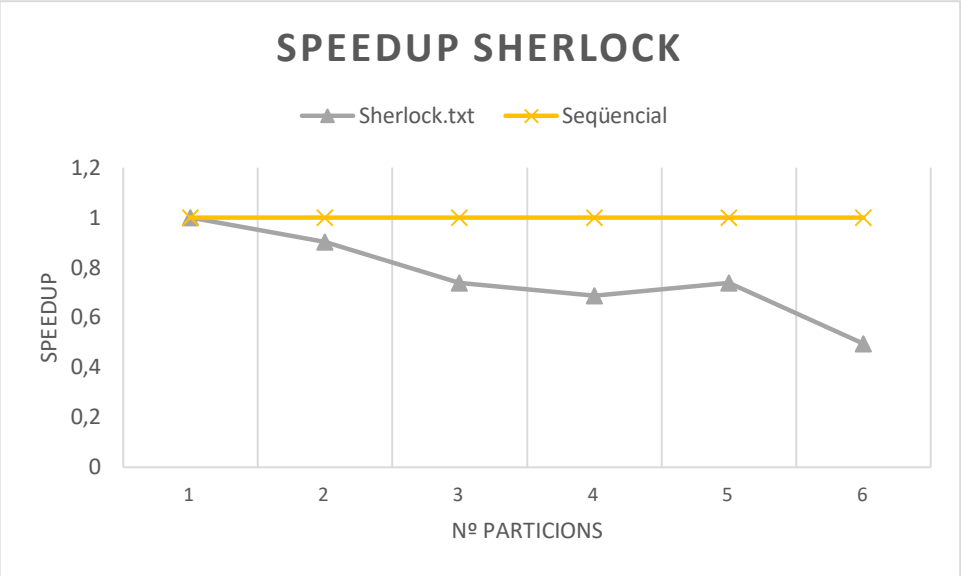
En aquest esperem a que el Reduce acabi. En aquest cas només hi ha un Count Words per tant esperem a que es creí el 'count' (fitxer que crea el Count Words quan acaba) al *cloud.*

4. Gràfiques speedup

4.1. Gràfiques individuals







A les tres gràfiques podem observar un comportament semblant. En les proves realitzades el *speedup* és inferior a '1' i a mida que incrementem les particions també disminueix el *speedup*. Aquest comportament és el contrari al esperat ja que normalment si incrementes les particions del treball el *speedup* augmenta de la mateixa manera.
Aquest comportament pot ser degut a diferents raons; La primera és que la mida dels fitxers utilitzats per realitzar les proves no són suficientment grans com per obtenir un increment del *speedup*. Un altra opció és que el mètode de Reduce no estigui optimitzat de tal manera que empitjori els temps i per tant el *speedup*.

4.2. Gràfica global