MBIT School
BUSINESS DATA INTELLIGENCE
We know how

Caso Práctico: MapReduce

M01-Ecosistema Big Data

DC02: Hadoop y Distribuciones Cloudera. El concepto de Data Lake

Programa: **Big Data, Cloud & Analytics**

Periodo académico: **19/20**

Autor/es: **Juan Ramón de Torre**

Antecedentes

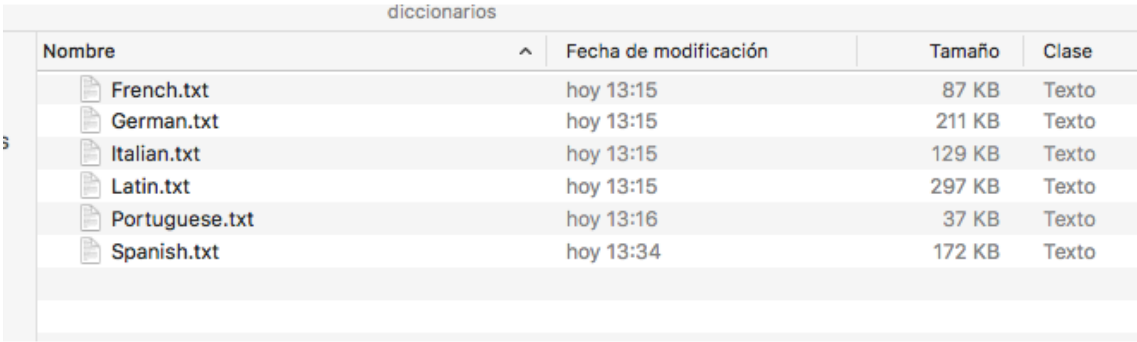
Los antecedentes descritos en el enunciado del caso práctico que debemos tomar como hipótesis son los siguientes:

- Debemos obtener un repositorio de términos para poder traducir a diferentes idiomas.
- Disponemos de unos diccionarios de inglés a diferentes idiomas.
- Cada fichero contiene términos y su traducción a un determinado idioma, separados por un tabulador.
- Para evitar complejidad: No nos importa si todos los términos figuran en todos los idiomas. Tampoco si un término tiene varias acepciones en un mismo idioma.

A continuación se describen los procesos seguidos para dar con el resultado al caso práctic.

Preprocesado de datos

Descargo todos los diccionarios en una carpeta local: /Users/hitoridekimasu/Documents/MasterBD/DC02-Hadoop_y_Distribuciones_Cloudera/Caso practico/diccionarios

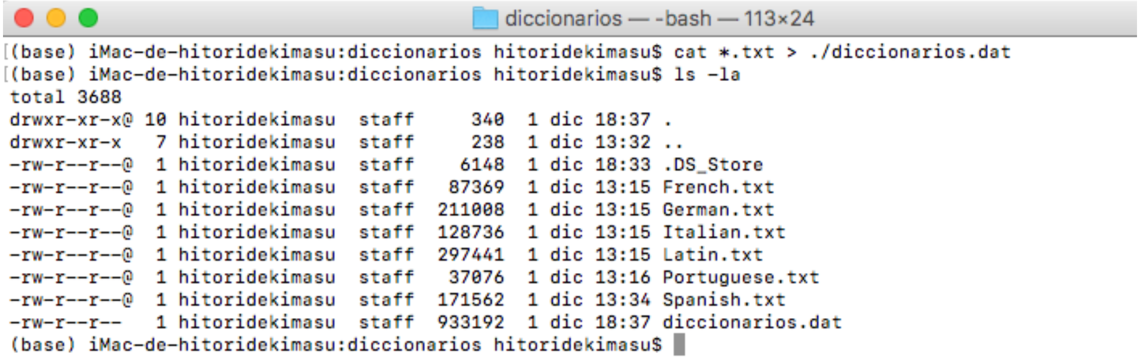


diccionarios

Nombre	Fecha de modificación	Tamaño	Clase
French.txt	hoy 13:15	87 KB	Texto
German.txt	hoy 13:15	211 KB	Texto
Italian.txt	hoy 13:15	129 KB	Texto
Latin.txt	hoy 13:15	297 KB	Texto
Portuguese.txt	hoy 13:16	37 KB	Texto
Spanish.txt	hoy 13:34	172 KB	Texto

Concateno todos los diccionarios en un archivo:

```
cat *.txt > ./diccionarios.dat
```



```
[(base) iMac-de-hitoridekimasu:diccionarios hitoridekimasu$ cat *.txt > ./diccionarios.dat
[(base) iMac-de-hitoridekimasu:diccionarios hitoridekimasu$ ls -la
total 3688
drwxr-xr-x@ 10 hitoridekimasu  staff     340  1 dic 18:37 .
drwxr-xr-x   7 hitoridekimasu  staff     238  1 dic 13:32 ..
-rw-r--r--@  1 hitoridekimasu  staff    6148  1 dic 18:33 .DS_Store
-rw-r--r--@  1 hitoridekimasu  staff   87369  1 dic 13:15 French.txt
-rw-r--r--@  1 hitoridekimasu  staff  211008  1 dic 13:15 German.txt
-rw-r--r--@  1 hitoridekimasu  staff  128736  1 dic 13:15 Italian.txt
-rw-r--r--@  1 hitoridekimasu  staff  297441  1 dic 13:15 Latin.txt
-rw-r--r--@  1 hitoridekimasu  staff   37076  1 dic 13:16 Portuguese.txt
-rw-r--r--@  1 hitoridekimasu  staff  171562  1 dic 13:34 Spanish.txt
-rw-r--r--   1 hitoridekimasu  staff  933192  1 dic 18:37 diccionarios.dat
(base) iMac-de-hitoridekimasu:diccionarios hitoridekimasu$
```

Modificación de mapper

Modificación de mapper para eliminar las líneas que empiezan por # y los textos entre corchetes y paréntesis:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import sys
import re


# input comes from STDIN (standard input)
terms=''
for line in sys.stdin:
    # Limpiamos espacios, buscamos el tabulador y separamos en 2 elementos (tupla)
    if (not re.search('^#',line)):
        line=re.sub('\[.*?]','',line)
        line=re.sub('\(.*?\)','',line)
        terms = line.strip().split('\t')
        # Volcamos la salida por consola
        print '\t'.join(terms)
```

Prueba en local

```
cat diccionarios.dat | python2 mapper.py | sort
```

Da problemas el sort al comparar líneas:

```
sort: string comparison failed: Illegal byte sequence

sort: Set LC_ALL='C' to work around the problem.

sort: The strings compared were `weekday\td\355a de la semana'
and `weekend\tfin de semana'.
```

Aplicamos LC_ALL='C':

```
cat diccionarios.dat | python2 mapper.py | LC_ALL='C' sort
```

Funciona correctamente

Aplicamos el reducer:

```
cat diccionarios.dat | python2 mapper.py | LC_ALL='C' sort |
python2 reducer.py > resultado.txt
```

Funcionamiento correcto

```
cat diccionarios.dat | python2 mapper.py | LC_ALL='C' sort |
python2 reducer.py | grep house
```

```
house CASA|CASITA|Haus|Rente|Unterkunft|casa|casa|casa|das
Haus|la casa|maison
```

El resultado no está ordenado por idioma, incluso, varias traducciones del mismo idioma aparecen desordenadas porque están en mayúscula y minúscula y el comando sort ordena por encima las que están en mayúscula.

```
cat diccionarios.dat | python2 mapper.py | LC_ALL='C' sort | grep
house
```

```
house    CASA
house    CASITA
house    Haus
house    Rente
house    Unterkunft
house    casa
house    casa
house    casa
house    das Haus
house    la casa
house    maison
```

Vamos a modificar el mapper y el reducer pasar todo a minúsculas y para a evitar que en el resultado aparezcan palabras repetidas.

Mapper: pasamos todo a minúsculas (lower)

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import sys
import re

# input comes from STDIN (standard input)
terms=''
for line in sys.stdin:

    if (not re.search('^#',line)):
        # print 'orig: '+line
        line=re.sub('\[.*?]','',line)
        line=re.sub('\(.*?\)','',line)
        terms = line.strip().split('\t')
        if len(terms)>1:
            terms[1]=terms[1].lower()
        # Volcamos la salida por consola
        print '\t'.join(terms)
```

Reducer: saltamos las palabras que ya están en la traducción repetidos

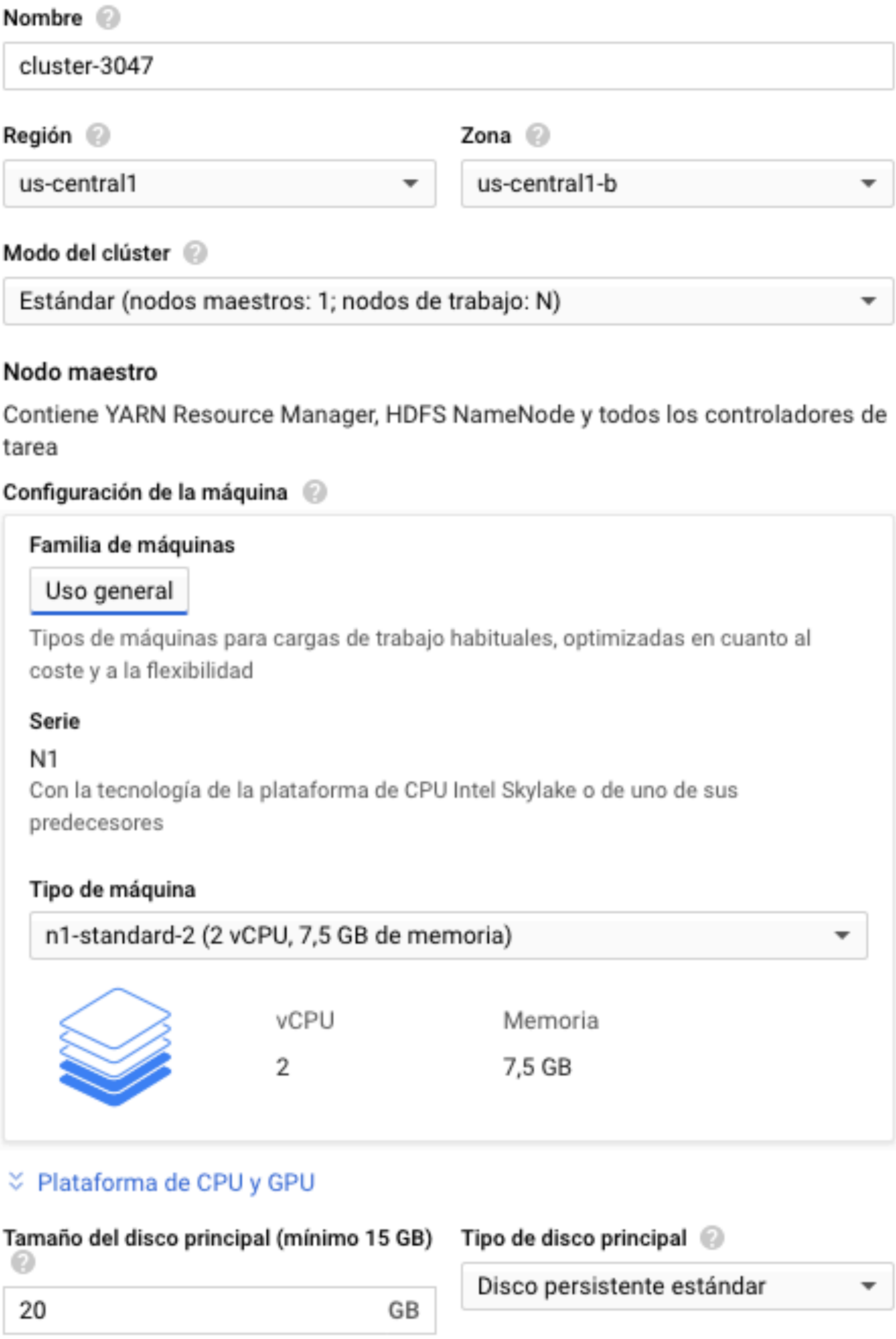
```
#!/usr/bin/env python
from operator import itemgetter
import sys
import re

current_word = None
word = None
trad_complete = None
# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip().split('\t')

    # parse the input we got from mapper.py
    word = line[0]
    try:
       trad = line[1]
    except:
       trad = ''
       pass
    if current_word == word:
        if trad_complete.find(trad)<0:
            trad_complete = trad_complete + "|" + trad
    else:
        if current_word:
            print '%s\t%s' % (current_word, trad_complete)
        trad_complete = trad
        current_word = word

# do not forget to output the last word if needed!
if current_word == word:
    print '%s\t%s' % (current_word, trad_complete)
```

Resultado:

```
house casa|casita|das haus|la casa|maison|rente|unterkunft
```

Una vez probado en local, lo vamos a ejecutar en Hadoop.

Levantamos Cluster



Nombre

cluster-3047

Región

us-central1

Zona

us-central1-b

Modo del clúster

Estándar (nodos maestros: 1; nodos de trabajo: N)

Nodo maestro

Contiene YARN Resource Manager, HDFS NameNode y todos los controladores de tarea

Configuración de la máquina

Familia de máquinas

Uso general

Tipos de máquinas para cargas de trabajo habituales, optimizadas en cuanto al coste y a la flexibilidad

Serie

N1

Con la tecnología de la plataforma de CPU Intel Skylake o de uno de sus predecesores

Tipo de máquina

n1-standard-2 (2 vCPU, 7,5 GB de memoria)

vCPU	Memoria
2	7,5 GB

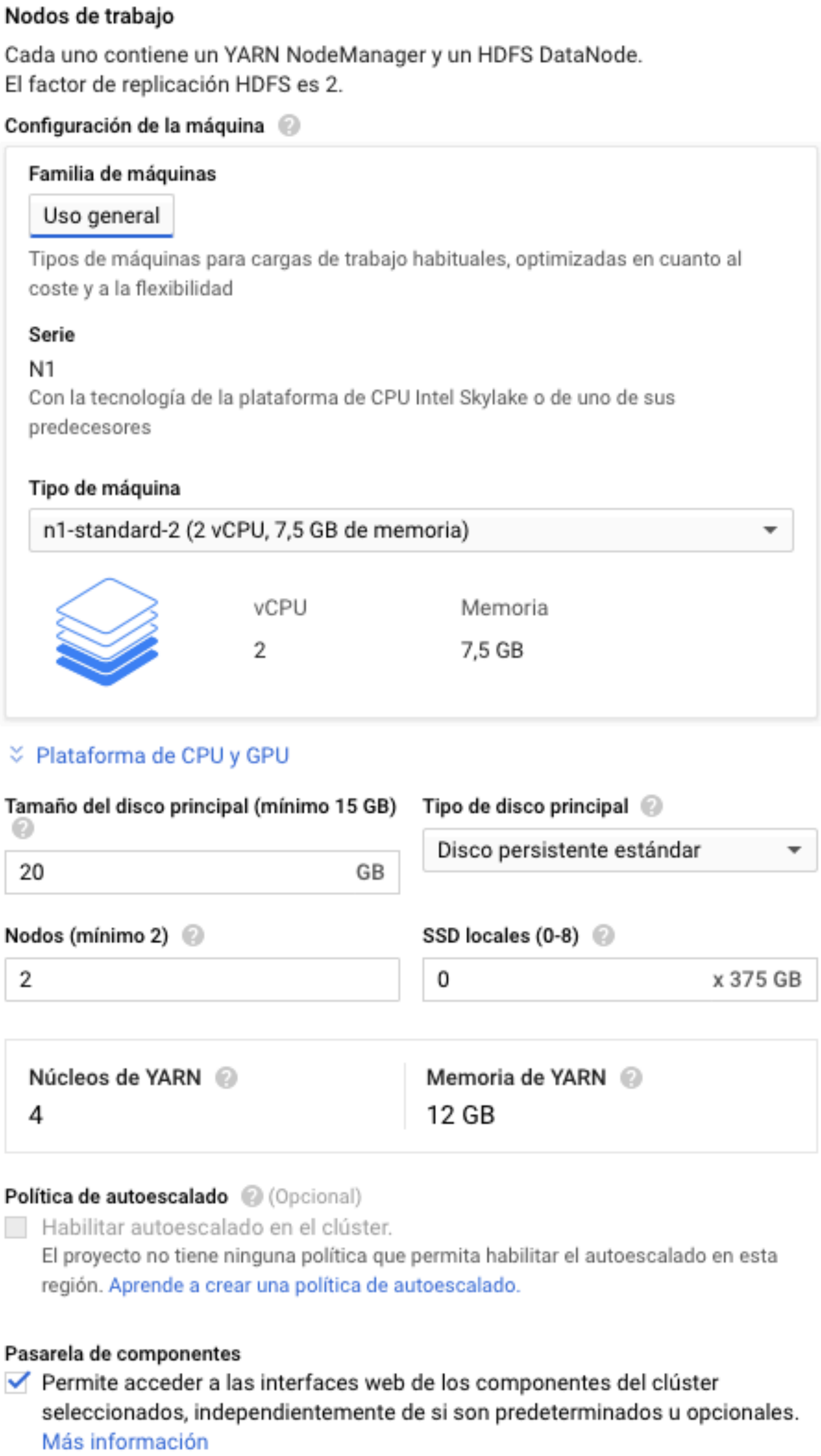
Plataforma de CPU y GPU

Tamaño del disco principal (mínimo 15 GB)

20 GB

Tipo de disco principal

Disco persistente estándar



Nodos de trabajo

Cada uno contiene un YARN NodeManager y un HDFS DataNode.
El factor de replicación HDFS es 2.

Configuración de la máquina

Familia de máquinas

Uso general

Tipos de máquinas para cargas de trabajo habituales, optimizadas en cuanto al coste y a la flexibilidad

Serie

N1

Con la tecnología de la plataforma de CPU Intel Skylake o de uno de sus predecesores

Tipo de máquina

n1-standard-2 (2 vCPU, 7,5 GB de memoria)

vCPU	Memoria
2	7,5 GB

Tamaño del disco principal (mínimo 15 GB): 20 GB

Tipo de disco principal: Disco persistente estándar

Nodos (mínimo 2): 2

SSD locales (0-8): 0 x 375 GB

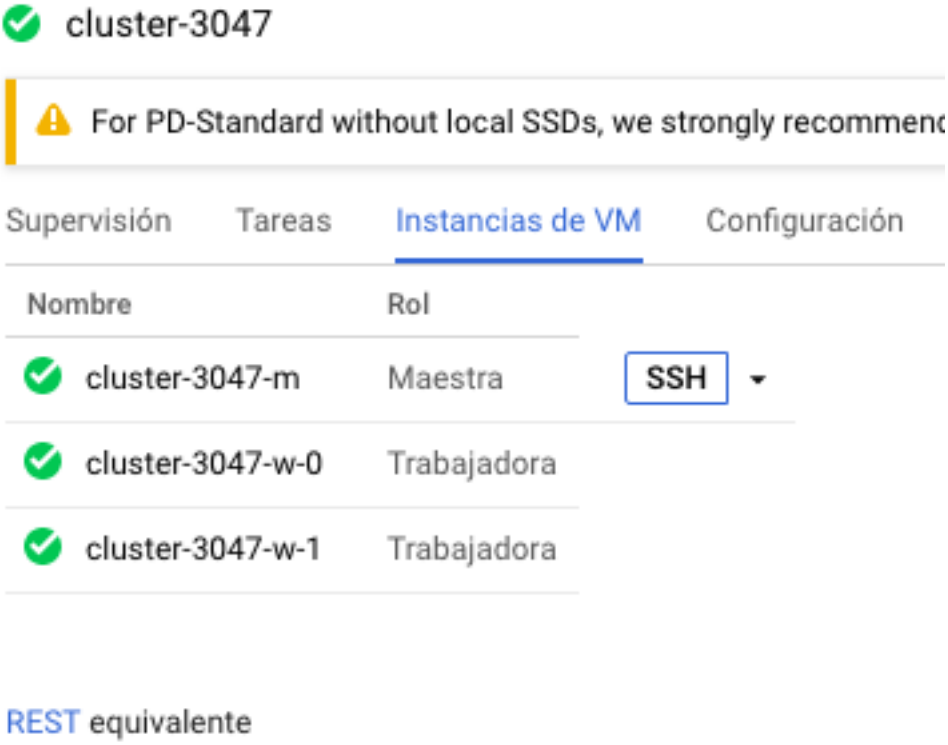
Núcleos de YARN	Memoria de YARN
4	12 GB

Política de autoescalado (Opcional)

□ Habilitar autoescalado en el clúster.
El proyecto no tiene ninguna política que permita habilitar el autoescalado en esta región. Aprende a crear una política de autoescalado.

Pasarela de componentes

☑ Permite acceder a las interfaces web de los componentes del clúster seleccionados, independientemente de si son predeterminados u opcionales.
Más información



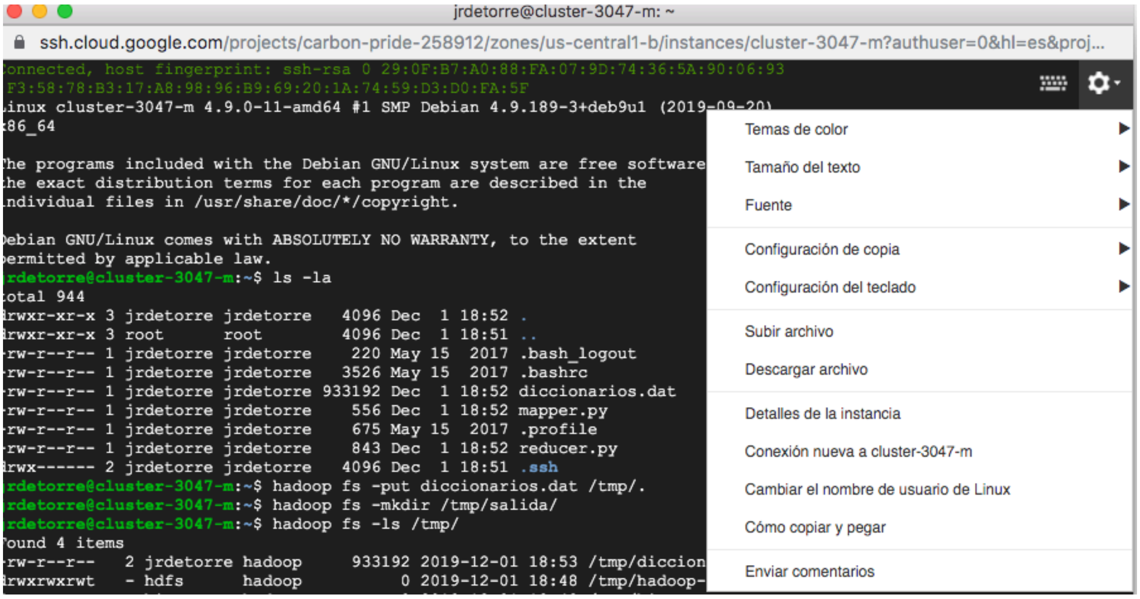
cluster-3047

For PD-Standard without local SSDs, we strongly recommenc

Nombre	Rol	
cluster-3047-m	Maestra	SSH ▾
cluster-3047-w-0	Trabajadora	
cluster-3047-w-1	Trabajadora	

Subimos archivos

Subimos los scripts y los diccionarios concatenados.

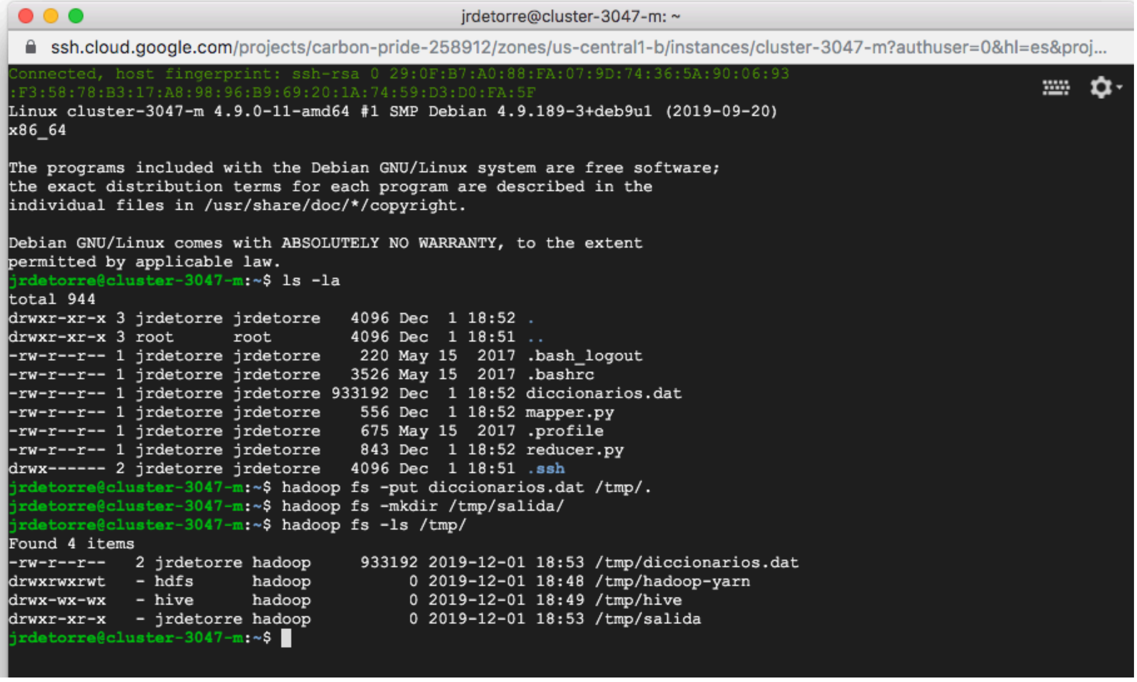


jrdetorre@cluster-3047-m: ~

ssh.cloud.google.com/projects/carbon-pride-258912/zones/us-central1-b/instances/cluster-3047-m?authuser=0&hl=es&proj...

```
onnected, host fingerprint: ssh-rsa 0 29:0F:B7:A0:88:FA:07:9D:74:36:5A:90:06:93
F3:58:78:B3:17:A8:98:96:B9:69:20:1A:74:59:D3:D0:FA:5F
inux cluster-3047-m 4.9.0-11-amd64 #1 SMP Debian 4.9.189-3+deb9u1 (2019-09-20)
86_64

he programs included with the Debian GNU/Linux system are free software
he exact distribution terms for each program are described in the
ndividual files in /usr/share/doc/*/copyright.

ebian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
ermitted by applicable law.
rdetorre@cluster-3047-m:~$ ls -la
otal 944
rwxr-xr-x 3 jrdetorre jrdetorre   4096 Dec  1 18:52 .
rwxr-xr-x 3 root      root        4096 Dec  1 18:51 ..
rw-r--r-- 1 jrdetorre jrdetorre    220 May 15  2017 .bash_logout
rw-r--r-- 1 jrdetorre jrdetorre   3526 May 15  2017 .bashrc
rw-r--r-- 1 jrdetorre jrdetorre 933192 Dec  1 18:52 diccionarios.dat
rw-r--r-- 1 jrdetorre jrdetorre    556 Dec  1 18:52 mapper.py
rw-r--r-- 1 jrdetorre jrdetorre    675 May 15  2017 .profile
rw-r--r-- 1 jrdetorre jrdetorre    843 Dec  1 18:52 reducer.py
rwx------ 2 jrdetorre jrdetorre   4096 Dec  1 18:51 .ssh
rdetorre@cluster-3047-m:~$ hadoop fs -put diccionarios.dat /tmp/.
rdetorre@cluster-3047-m:~$ hadoop fs -mkdir /tmp/salida/
rdetorre@cluster-3047-m:~$ hadoop fs -ls /tmp/
ound 4 items
rw-r--r--   2 jrdetorre hadoop     933192 2019-12-01 18:53 /tmp/diccion
rwxrwxrwt   - hdfs      hadoop          0 2019-12-01 18:48 /tmp/hadoop-
```

Creamos la carpeta de salida de resultados MapReduce.

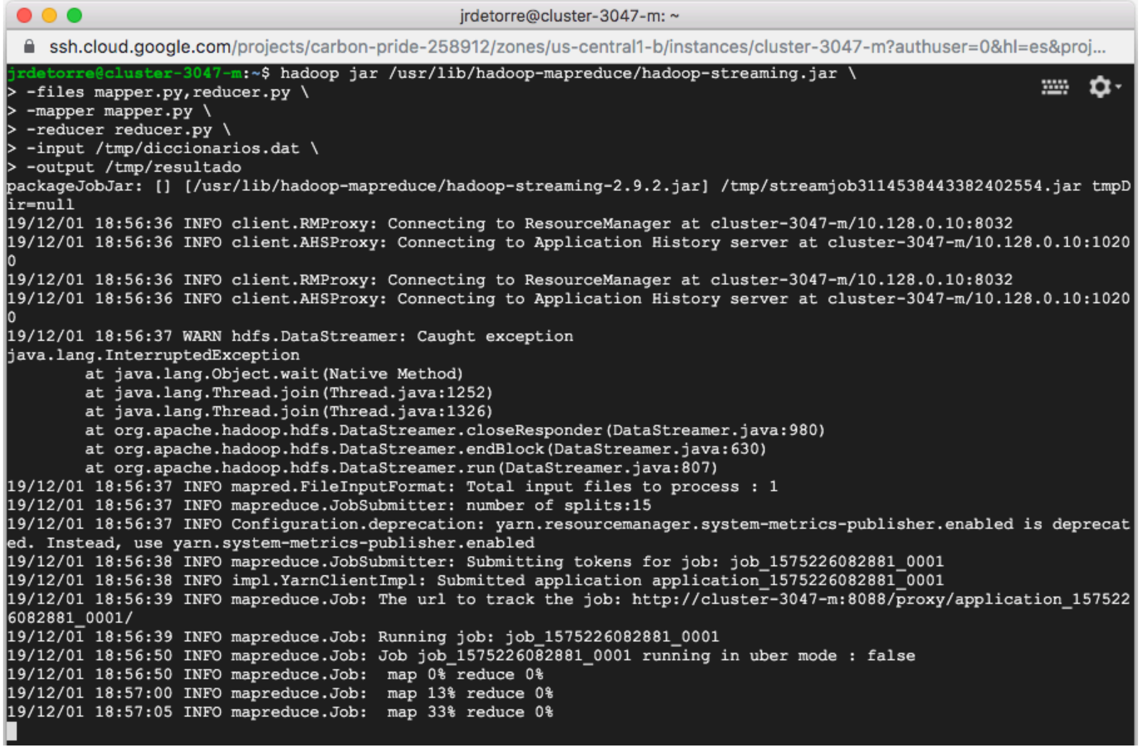

jrdetorre@cluster-3047-m: ~
ssh.cloud.google.com/projects/carbon-pride-258912/zones/us-central1-b/instances/cluster-3047-m?authuser=0&hl=es&proj...

```
Connected, host fingerprint: ssh-rsa 0 29:0F:B7:A0:88:FA:07:9D:74:36:5A:90:06:93
:F3:58:78:B3:17:A8:98:96:B9:69:20:1A:74:59:D3:D0:FA:5F
Linux cluster-3047-m 4.9.0-11-amd64 #1 SMP Debian 4.9.189-3+deb9u1 (2019-09-20)
x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
jrdetorre@cluster-3047-m:~$ ls -la
total 944
drwxr-xr-x 3 jrdetorre jrdetorre   4096 Dec  1 18:52 .
drwxr-xr-x 3 root      root        4096 Dec  1 18:51 ..
-rw-r--r-- 1 jrdetorre jrdetorre    220 May 15  2017 .bash_logout
-rw-r--r-- 1 jrdetorre jrdetorre   3526 May 15  2017 .bashrc
-rw-r--r-- 1 jrdetorre jrdetorre 933192 Dec  1 18:52 diccionarios.dat
-rw-r--r-- 1 jrdetorre jrdetorre    556 Dec  1 18:52 mapper.py
-rw-r--r-- 1 jrdetorre jrdetorre    675 May 15  2017 .profile
-rw-r--r-- 1 jrdetorre jrdetorre    843 Dec  1 18:52 reducer.py
drwx------ 2 jrdetorre jrdetorre   4096 Dec  1 18:51 .ssh
jrdetorre@cluster-3047-m:~$ hadoop fs -put diccionarios.dat /tmp/.
jrdetorre@cluster-3047-m:~$ hadoop fs -mkdir /tmp/salida/
jrdetorre@cluster-3047-m:~$ hadoop fs -ls /tmp/
Found 4 items
-rw-r--r--   2 jrdetorre hadoop     933192 2019-12-01 18:53 /tmp/diccionarios.dat
drwxrwxrwt   - hdfs      hadoop          0 2019-12-01 18:48 /tmp/hadoop-yarn
drwx-wx-wx   - hive      hadoop          0 2019-12-01 18:49 /tmp/hive
drwxr-xr-x   - jrdetorre hadoop          0 2019-12-01 18:53 /tmp/salida
jrdetorre@cluster-3047-m:~$ █
```

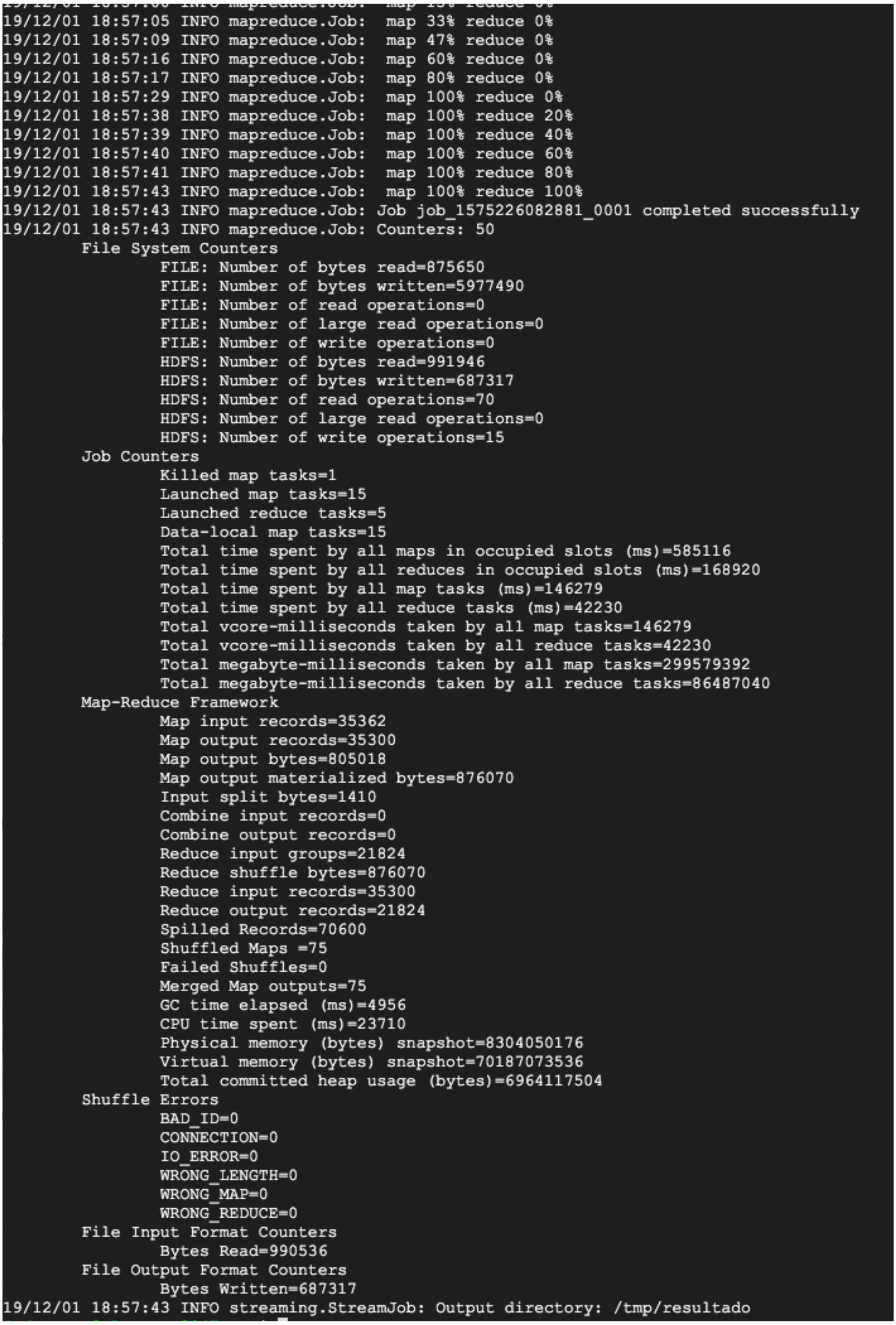
Comando Hadoop - MapReduce

Ejecutamos el siguiente comando.

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar \

-files mapper.py,reducer.py \

-mapper mapper.py \

-reducer reducer.py \

-input /tmp/diccionarios.dat \

-output /tmp/resultado
```

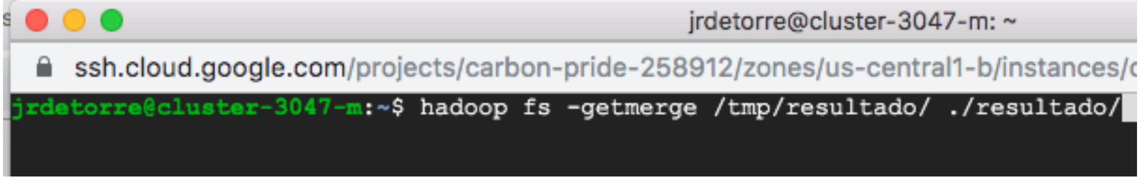

jrdetorre@cluster-3047-m: ~
ssh.cloud.google.com/projects/carbon-pride-258912/zones/us-central1-b/instances/cluster-3047-m?authuser=0&hl=es&proj...

```
jrdetorre@cluster-3047-m:~$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar \
> -files mapper.py,reducer.py \
> -mapper mapper.py \
> -reducer reducer.py \
> -input /tmp/diccionarios.dat \
> -output /tmp/resultado
packageJobJar: [] [/usr/lib/hadoop-mapreduce/hadoop-streaming-2.9.2.jar] /tmp/streamjob3114538443382402554.jar tmpD
ir=null
19/12/01 18:56:36 INFO client.RMProxy: Connecting to ResourceManager at cluster-3047-m/10.128.0.10:8032
19/12/01 18:56:36 INFO client.AHSProxy: Connecting to Application History server at cluster-3047-m/10.128.0.10:1020
0
19/12/01 18:56:36 INFO client.RMProxy: Connecting to ResourceManager at cluster-3047-m/10.128.0.10:8032
19/12/01 18:56:36 INFO client.AHSProxy: Connecting to Application History server at cluster-3047-m/10.128.0.10:1020
0
19/12/01 18:56:37 WARN hdfs.DataStreamer: Caught exception
java.lang.InterruptedException
        at java.lang.Object.wait(Native Method)
        at java.lang.Thread.join(Thread.java:1252)
        at java.lang.Thread.join(Thread.java:1326)
        at org.apache.hadoop.hdfs.DataStreamer.closeResponder(DataStreamer.java:980)
        at org.apache.hadoop.hdfs.DataStreamer.endBlock(DataStreamer.java:630)
        at org.apache.hadoop.hdfs.DataStreamer.run(DataStreamer.java:807)
19/12/01 18:56:37 INFO mapred.FileInputFormat: Total input files to process : 1
19/12/01 18:56:37 INFO mapreduce.JobSubmitter: number of splits:15
19/12/01 18:56:37 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecat
ed. Instead, use yarn.system-metrics-publisher.enabled
19/12/01 18:56:38 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1575226082881_0001
19/12/01 18:56:38 INFO impl.YarnClientImpl: Submitted application application_1575226082881_0001
19/12/01 18:56:39 INFO mapreduce.Job: The url to track the job: http://cluster-3047-m:8088/proxy/application_157522
6082881_0001/
19/12/01 18:56:39 INFO mapreduce.Job: Running job: job_1575226082881_0001
19/12/01 18:56:50 INFO mapreduce.Job: Job job_1575226082881_0001 running in uber mode : false
19/12/01 18:56:50 INFO mapreduce.Job:  map 0% reduce 0%
19/12/01 18:57:00 INFO mapreduce.Job:  map 13% reduce 0%
19/12/01 18:57:05 INFO mapreduce.Job:  map 33% reduce 0%
```



```
19/12/01 18:57:00 INFO mapreduce.Job:  map 13% reduce 0%
19/12/01 18:57:05 INFO mapreduce.Job:  map 33% reduce 0%
19/12/01 18:57:09 INFO mapreduce.Job:  map 47% reduce 0%
19/12/01 18:57:16 INFO mapreduce.Job:  map 60% reduce 0%
19/12/01 18:57:17 INFO mapreduce.Job:  map 80% reduce 0%
19/12/01 18:57:29 INFO mapreduce.Job:  map 100% reduce 0%
19/12/01 18:57:38 INFO mapreduce.Job:  map 100% reduce 20%
19/12/01 18:57:39 INFO mapreduce.Job:  map 100% reduce 40%
19/12/01 18:57:40 INFO mapreduce.Job:  map 100% reduce 60%
19/12/01 18:57:41 INFO mapreduce.Job:  map 100% reduce 80%
19/12/01 18:57:43 INFO mapreduce.Job:  map 100% reduce 100%
19/12/01 18:57:43 INFO mapreduce.Job: Job job_1575226082881_0001 completed successfully
19/12/01 18:57:43 INFO mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=875650
                FILE: Number of bytes written=5977490
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=991946
                HDFS: Number of bytes written=687317
                HDFS: Number of read operations=70
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=15
        Job Counters
                Killed map tasks=1
                Launched map tasks=15
                Launched reduce tasks=5
                Data-local map tasks=15
                Total time spent by all maps in occupied slots (ms)=585116
                Total time spent by all reduces in occupied slots (ms)=168920
                Total time spent by all map tasks (ms)=146279
                Total time spent by all reduce tasks (ms)=42230
                Total vcore-milliseconds taken by all map tasks=146279
                Total vcore-milliseconds taken by all reduce tasks=42230
                Total megabyte-milliseconds taken by all map tasks=299579392
                Total megabyte-milliseconds taken by all reduce tasks=86487040
        Map-Reduce Framework
                Map input records=35362
                Map output records=35300
                Map output bytes=805018
                Map output materialized bytes=876070
                Input split bytes=1410
                Combine input records=0
                Combine output records=0
                Reduce input groups=21824
                Reduce shuffle bytes=876070
                Reduce input records=35300
                Reduce output records=21824
                Spilled Records=70600
                Shuffled Maps =75
                Failed Shuffles=0
                Merged Map outputs=75
                GC time elapsed (ms)=4956
                CPU time spent (ms)=23710
                Physical memory (bytes) snapshot=8304050176
                Virtual memory (bytes) snapshot=70187073536
                Total committed heap usage (bytes)=6964117504
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=990536
        File Output Format Counters
                Bytes Written=687317
19/12/01 18:57:43 INFO streaming.StreamJob: Output directory: /tmp/resultado
```

Al haber utilizado 6 tareas de Reduce, tenemos que combinar los resultados de salida.


jrdetorre@cluster-3047-m: ~
ssh.cloud.google.com/projects/carbon-pride-258912/zones/us-central1-b/instances/c

```
jrdetorre@cluster-3047-m:~$ hadoop fs -getmerge /tmp/resultado/ ./resultado/
```

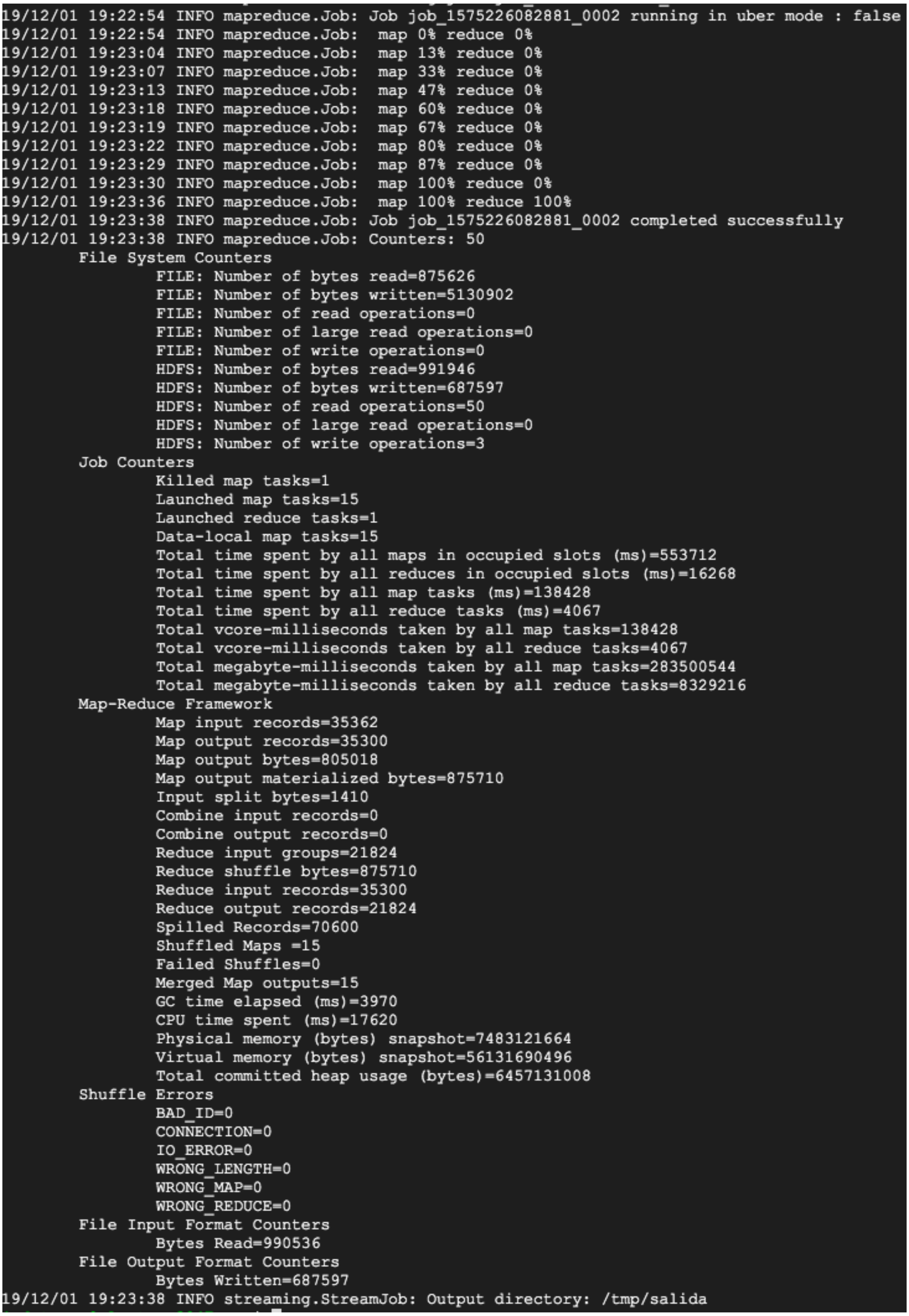
```
hadoop fs -getmerge /tmp/resultado/ ./resultado/
```

Resultado:

```
house casa|casita|la casa|haus|das haus|rente|unterkunft|maison
```

El resultado es correcto, pero las partes de cada reducer no se han unido ordenadamente. Vamos a intentar utilizar un único reducer para que la salida esté ordenada.

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar \
-D mapred.reduce.tasks=1 \
-files mapper.py,reducer.py \
-mapper mapper.py \
-reducer reducer.py \
-input /tmp/diccionarios.dat \
-output /tmp/salida
```



```
19/12/01 19:22:54 INFO mapreduce.Job: Job job_1575226082881_0002 running in uber mode : false
19/12/01 19:22:54 INFO mapreduce.Job:  map 0% reduce 0%
19/12/01 19:23:04 INFO mapreduce.Job:  map 13% reduce 0%
19/12/01 19:23:07 INFO mapreduce.Job:  map 33% reduce 0%
19/12/01 19:23:13 INFO mapreduce.Job:  map 47% reduce 0%
19/12/01 19:23:18 INFO mapreduce.Job:  map 60% reduce 0%
19/12/01 19:23:19 INFO mapreduce.Job:  map 67% reduce 0%
19/12/01 19:23:22 INFO mapreduce.Job:  map 80% reduce 0%
19/12/01 19:23:29 INFO mapreduce.Job:  map 87% reduce 0%
19/12/01 19:23:30 INFO mapreduce.Job:  map 100% reduce 0%
19/12/01 19:23:36 INFO mapreduce.Job:  map 100% reduce 100%
19/12/01 19:23:38 INFO mapreduce.Job: Job job_1575226082881_0002 completed successfully
19/12/01 19:23:38 INFO mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=875626
                FILE: Number of bytes written=5130902
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=991946
                HDFS: Number of bytes written=687597
                HDFS: Number of read operations=50
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=3
        Job Counters
                Killed map tasks=1
                Launched map tasks=15
                Launched reduce tasks=1
                Data-local map tasks=15
                Total time spent by all maps in occupied slots (ms)=553712
                Total time spent by all reduces in occupied slots (ms)=16268
                Total time spent by all map tasks (ms)=138428
                Total time spent by all reduce tasks (ms)=4067
                Total vcore-milliseconds taken by all map tasks=138428
                Total vcore-milliseconds taken by all reduce tasks=4067
                Total megabyte-milliseconds taken by all map tasks=283500544
                Total megabyte-milliseconds taken by all reduce tasks=8329216
        Map-Reduce Framework
                Map input records=35362
                Map output records=35300
                Map output bytes=805018
                Map output materialized bytes=875710
                Input split bytes=1410
                Combine input records=0
                Combine output records=0
                Reduce input groups=21824
                Reduce shuffle bytes=875710
                Reduce input records=35300
                Reduce output records=21824
                Spilled Records=70600
                Shuffled Maps =15
                Failed Shuffles=0
                Merged Map outputs=15
                GC time elapsed (ms)=3970
                CPU time spent (ms)=17620
                Physical memory (bytes) snapshot=7483121664
                Virtual memory (bytes) snapshot=56131690496
                Total committed heap usage (bytes)=6457131008
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=990536
        File Output Format Counters
                Bytes Written=687597
19/12/01 19:23:38 INFO streaming.StreamJob: Output directory: /tmp/salida
```

```
hadoop fs -getmerge /tmp/salida/ ./salida.txt
```

Resultado final

```
cat salida.txt | grep house
```

```
house casita|casa|la casa|maison|rente|das haus|unterkunft
```