

# Earthquake Application

## Guida all'uso



```
## 🏠 PARTE 1: Test in Locale
```

```
### Prerequisiti
- Java 17
- Scala 2.13.x
- SBT
- Apache Spark 4.0.1 (per test locali)
- Google Cloud SDK (per DataProc)
```

```
```bash
# Per verificare di avere tutto installato
java -version      # Deve essere 17 o superiore
scala -version     # Deve essere 2.13.x
sbt --version       # Qualsiasi versione recente
spark-submit --version # Deve essere 4.x
```
```

```
### Step 1: Compila il Progetto
```

```
```bash
cd /path/to/your/project

# Compila
sbt clean compile

# Crea il JAR
sbt assembly
```
```

Il JAR verrà creato in: `target/scala-2.13/earthquake-cooccurrence-assembly-1.0.jar`

```
### Step 2: Esegui Test Locale
```

```
```bash
# Test con approccio 1 (GroupByKey) e Hash partitioner
spark-submit \
    --class Main \
    --master local[*] \
    target/scala-2.13/earthquake-cooccurrence-assembly-1.0.jar \
    test-data.csv \
    output-local-test \
    4 \
    groupbykey \
    hash \
    1

# Vedi i risultati
cat output-test/metrics-readable/part-*
```
```

```
### Step 3: Test Tutti gli Approcci in Locale
```

```
```bash
# Script automatico per testare tutto
chmod +x test-all-approaches.sh
./test-all-approaches.sh
```
```

Questo testerà:

- GroupByKey
- AggregateByKey
- ReduceByKey

E verificherà che producano tutti lo stesso risultato.

---

## 🖱 PARTE 2: Esecuzione su Google Cloud

### Prerequisiti Google Cloud

```
```bash
# Installa Google Cloud SDK (se non l'hai già)
# Visita: https://cloud.google.com/sdk/docs/install

# Fai login
gcloud auth login

# Imposta il progetto (usa il TUO project ID)
gcloud config set project YOUR_PROJECT_ID

# Verifica
gcloud config list
```
```

### Step 1: Crea un Bucket su Google Cloud Storage

```
```bash
# Scegli un nome univoco per il bucket
BUCKET_NAME="earthquake-analysis-TUOMATRICOLA"

# Crea il bucket
gsutil mb gs://$BUCKET_NAME/

# Verifica
gsutil ls
```
```

### Step 2: Upload del JAR e Dataset

```
```bash
# Upload JAR
gsutil cp target/scala-2.12/earthquake-cooccurrence-assembly-1.0.jar \
gs://$BUCKET_NAME/jars/

# Upload del dataset (usa il TUO file!)
gsutil cp /path/to/earthquakes-full.csv \
gs://$BUCKET_NAME/data/

# Verifica upload
gsutil ls gs://$BUCKET_NAME/jars/
gsutil ls gs://$BUCKET_NAME/data/
```
```

### Step 3: Crea un Cluster DataProc

```
```bash
# Cluster con 2 workers
gcloud dataproc clusters create earthquake-cluster-2w \
--region=europe-west1 \
--num-workers 2 \
--master-boot-disk-size 240 \
--worker-boot-disk-size 240 \
--master-machine-type=n2-standard-4 \
--worker-machine-type=n2-standard-4

# IMPORTANTE: Aspetta che il cluster sia pronto (1-2 minuti)
```

```

### Step 4: Esegui UN Job Singolo

```
```bash
# Parametri
BUCKET_NAME="earthquake-analysis-TUOMATRICOLA" # Il TUO bucket!
CLUSTER_NAME="earthquake-cluster-2w"
REGION="europe-west1"

# Esegui job con GroupByKey e Hash partitioner
gcloud dataproc jobs submit spark \
--cluster=$CLUSTER_NAME \
--region=$REGION \
--jar=gs://$BUCKET_NAME/jars/earthquake-cooccurrence-assembly-1.0.jar \
-- gs://$BUCKET_NAME/data/earthquakes-full.csv \
  gs://$BUCKET_NAME/output/test-run \
    8 \
      groupbykey \
        hash \
          2

```

# Attendi che il job finisca (guarda nella console)

### Step 5: Scarica i Risultati

```
```bash
# Scarica output
gsutil cp -r gs://$BUCKET_NAME/output/test-run ./results-test/

# Vedi il risultato
cat results-test/part-*

# Vedi le metriche
cat results-test/metrics/part-*
```

```

### Step 6: Elimina il Cluster (IMPORTANTE!)

```
```bash
# Elimina il cluster per non consumare crediti
gcloud dataproc clusters delete earthquake-cluster-2w \
--region=europe-west1
```

```

---

```
## 💡 PARTE 3: Esperimenti Completati (Per il Report)

### Opzione A: Script Automatico (RACCOMANDATO)

```bash
chmod +x run-complete-experiments.sh

# Questo eseguirà TUTTI i 18 esperimenti:
# - 3 approcci × 2 partitioners × 3 configurazioni workers
./run-complete-experiments.sh YOUR_BUCKET_NAME earthquakes-full.csv
```

```

Lo script farà TUTTO automaticamente:

1.  Crea cluster 2 workers → esegue 6 esperimenti → elimina cluster
2.  Crea cluster 3 workers → esegue 6 esperimenti → elimina cluster
3.  Crea cluster 4 workers → esegue 6 esperimenti → elimina cluster
4.  Genera CSV finale con TUTTE le metriche
5.  Genera report con calcoli automatici

### Opzione B: Manuale (Un Esperimento alla Volta)

Se preferisci controllare ogni step:

```
```bash
# 1. Crea cluster
gcloud dataproc clusters create earthquake-cluster-2w \
    --region=europe-west1 \
    --num-workers 2 \
    --master-boot-disk-size 240 \
    --worker-boot-disk-size 240 \
    --master-machine-type=n2-standard-4 \
    --worker-machine-type=n2-standard-4

# 2. Esegui esperimenti (cambia approach e partitioner)
for approach in groupbykey aggregatebykey reducebykey; do
    for part in hash range; do
        gcloud dataproc jobs submit spark \
            --cluster=earthquake-cluster-2w \
            --region=europe-west1 \
            --jar=gs://YOUR_BUCKET/jars/earthquake-cooccurrence-assembly-
1.0.jar \
            -- gs://YOUR_BUCKET/data/earthquakes-full.csv \
            gs://YOUR_BUCKET/output/2w-${approach}-${part} \
            8 \
            $approach \
            $part \
            2

        # Scarica metriche
        gsutil cp gs://YOUR_BUCKET/output/2w-${approach}-
${part}/metrics/part-* \
            metrics-2w-${approach}-${part}.csv
    done
done

# 3. Elimina cluster
gcloud dataproc clusters delete earthquake-cluster-2w --region=europe-
west1
```

```

```
# 4. Ripeti per 3 e 4 workers
` `` `
```

---

```
## 📈 Interpretare le Metriche
```

Il file CSV generato avrà queste colonne:

```
```csv
approach,partitioner,num_workers,num_partitions,total_events,unique_event
s,co_occurrences,load_time_ms,analysis_time_ms,total_time_ms,max_count,ti
mestamp
GroupByKey,Hash,2,8,1000000,950000,50000,12345,45678,58023,150,1234567890
````
```

### Metriche Principali:

1. \*\*total\_time\_ms\*\*: Tempo totale (questo è il più importante!)

2. \*\*Speedup\*\*:

```
```
S(n) = T(baseline) / T(current)
dove baseline = GroupByKey-Hash con 2 workers
````
```

3. \*\*Strong Scaling Efficiency\*\*:

```
```
E(n) = T(2) / (n × T(n) / 2)
````
```

4. \*\*Confronto Partitioner\*\*:

```
```
Differenza % = (T_hash - T_range) / T_hash × 100
````
```

---

```
## 🔎 Configurazioni da Testare
```

### MINIMO (Per consegna base):

````

- ✓ 2 workers: GroupByKey-Hash
- ✓ 3 workers: GroupByKey-Hash
- ✓ 4 workers: GroupByKey-Hash

### RACCOMANDATO (Per buon voto):

````

- ✓ Tutti e 3 gli approcci
- ✓ Hash e Range partitioner
- ✓ 2, 3, 4 workers  
= 18 esperimenti totali

---

## ## ! COSE IMPORTANTI

```
### 1. Sempre Eliminare i Cluster!
```bash
# Lista cluster attivi
gcloud dataproc clusters list --region=europe-west1

# Elimina TUTTI
gcloud dataproc clusters delete CLUSTER_NAME --region=europe-west1
```
```

## ### 2. Controlla i Costi

```
```bash
# Vedi quanto stai spendendo
gcloud billing accounts list
# Vai su: https://console.cloud.google.com/billing
```
```

## ### 3. Dataset Corretto

- ✗ NON usare i dati di esempio dalla traccia!
- ✓ USA il file `earthquakes-full.csv` fornito dal prof
- ✓ O un dataset reale di terremoti

## ### 4. Formato Output

Il risultato REALE dipenderà dal TUO dataset. Formato:

```

```
((lat1, lon1), (lat2, lon2))
data1
data2
data3
:::
```

---

## ## 🛡 Problemi Comuni

```
### "Permission denied" su Google Cloud
```bash
gcloud auth login
gcloud auth application-default login
```
```

```
### "Cluster creation failed"
- Verifica di avere crediti education attivi
- Prova una regione diversa (es. `us-central1`)
- Verifica quota workers nel tuo progetto
```

```
### "OutOfMemoryError"
- Aumenta `--driver-memory` e `--executor-memory`
- Usa AggregateByKey invece di GroupByKey
```

```
### Job troppo lento
- Aumenta numero di partizioni
- Usa AggregateByKey
- Verifica di non usare troppi workers (overhead)
```

---

## ## 📈 Per il Report

Dopo aver eseguito gli esperimenti:

1.  Apri `final\_metrics\_\*.csv` in Excel/Google Sheets
2.  Crea tavole pivot per raggruppare dati
3.  Genera grafici:
  - Tempo vs Workers (linee)
  - Speedup vs Workers (linee)
  - Confronto approcci (barre)
  - Hash vs Range (barre)
4.  Calcola metriche con formule Excel
5.  Spiega i risultati osservati

---

## ##💡 Consigli Finali

1. \*\*Inizia SEMPLICE\*\*: Prima un test locale, poi uno su Cloud
2. \*\*Un passo alla volta\*\*: Non lanciare tutti gli esperimenti insieme
3. \*\*Verifica sempre\*\*: Controlla che i risultati siano sensati
4. \*\*Documenta tutto\*\*: Salva log e screenshot per il report
5. \*\*Elimina cluster\*\*: SEMPRE eliminare dopo ogni uso

---

## ##🚀 Quick Start TL;DR

```
```bash
# LOCALE
sbt assembly
spark-submit --class Main --master local[*] \
  target/scala-2.12/earthquake-cooccurrence-assembly-1.0.jar \
  test-data.csv output 4 aggregateByKey hash 1

# CLOUD (completo automatico)
./run-complete-experiments.sh YOUR_BUCKET earthquakes-full.csv

# Risultati
cat final_metrics_*.csv
````
```

Fatto! 🎉