Running a hadoop job on GCP

how to think about hadoop

Hadoop runs map reduce by streaming data over stdin and stdout. Therefore, your map and reduce scripts need to read data in from the standard input (`sys.stdin`) and write their results to the standard output (`print` functions).

Think of a hadoop job running in 3 phases:

phase 1: map

in parallel, worker nodes read pieces of the full dataset from stdin and print out a series of key-value pairs. These keys/values can be anything. Use the face that the keys will be sorted to your advantage

phase 2: sort / partition

After worker nodes generate the key-value pairs, the master node will sort the key-value pairs by key. It will then send these sorted key-value pairs to worker nodes. All the data belonging to one key will go to the same worker node.

phase 3: reduce

worker nodes process the key-value pairs that they recieved on a per-key basis, writing their results to the stdout

writing the map-reduce scripts

When debugging hadoop jobs on GCP, I noticed that a lot of errors may have been caused by improper shebangs. For python scripts, I used `#!/usr/bin/env python3` and `#!/usr/bin/bash` for bash scripts.

mapper

The job of the mapper is to read a bunch of lines of text from the standard input and print key-value pairs to the standard output. A template to start with is:

```
#!/usr/bin/env python3

import sys
```

```
for line in sys.stdin:

        # do some processing

        # split the line into tokens

        # send the key-value pair to standard output
        print(f"{key},{value}")
```

reducer

The job of the reducer is to read the key-value pairs generated by the mapper from the stdin.

The job of the reducer is to read key-value pairs from standard input. The key-value pairs recieved will all be of the same key and all keys of that type will be on the same worker. Here's a template to start with:

```
#!/usr/bin/env python3

import sys

for line in sys.stdin:
        key, value = line.split(',') # you can use any delimiter you want
        # do some calculations

# print the final result to standard output
print(f"final result for key {key} is: {result}"")
```

testing the map-reduce scripts

You can simulate running a hadoop job locally using this bash command:

```
cat input_dataset.txt | python mapper.py | sort | python reducer.py > output.txt
```

setting up dataproc

in progress

submitting the job

you can use the following command on a dataproc cluster to run a job:

```
hadoop jar /usr/lib/hadoop/hadoop-streaming.jar \
        -files mapper.py,reducer.py \
        -mapper mapper.py \
        -reducer reducer.py \
        -input input_data.txt \
        -output output_file.txt
```

to collect the output files from the hdfs:

```
hdfs dfs -getmerge output_file.txt/part* output_file.txt
```

processing the results

References

- https://www.databricks.com/glossary/mapreduce
- https://stackoverflow.com/questions/28982/simple-explanation-of-mapreduce
- https://en.wikipedia.org/wiki/MapReduce
- numerous conversations with chatGPT