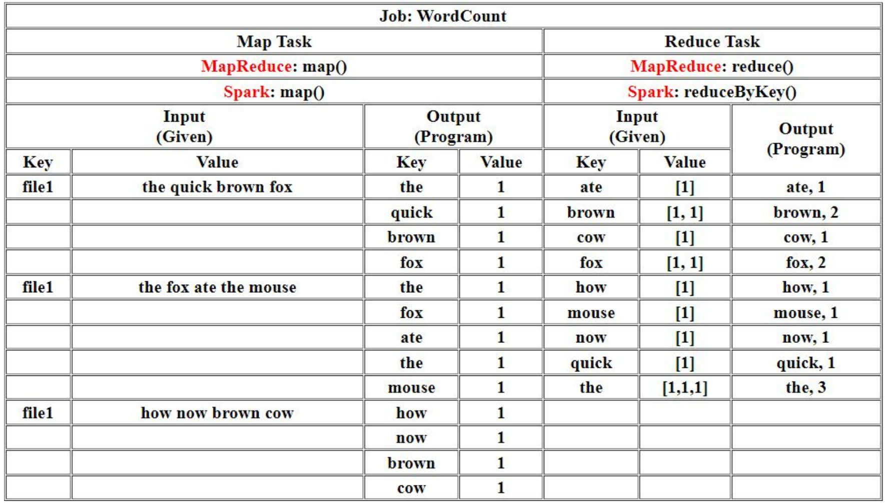
PySpark on Kubernetes

CS570: Big Data Processing & Analytics

Project by: Imran N Saleh Student ID 19648

Introduction: Word Count

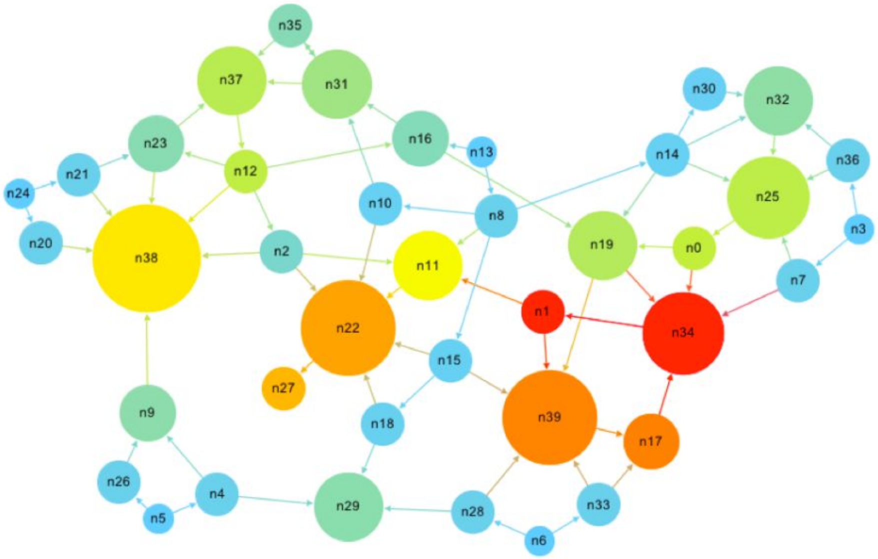
Word Count: The word count algorithm involves counting the occurrences of each unique word in a given text or document. With Spark's parallel processing capabilities, the word count operation can be distributed across multiple nodes in the Kubernetes cluster for faster and efficient execution.



Job: WordCount						
Map Task				Reduce Task		
MapReduce: map()				MapReduce: reduce()		
Spark: map()				Spark: reduceByKey()		
Input (Given)		Output (Program)		Input (Given)		Output (Program)
Key	Value	Key	Value	Key	Value	
file1	the quick brown fox	the	1	ate	[1]	ate, 1
		quick	1	brown	[1, 1]	brown, 2
		brown	1	cow	[1]	cow, 1
		fox	1	fox	[1, 1]	fox, 2
file1	the fox ate the mouse	the	1	how	[1]	how, 1
		fox	1	mouse	[1]	mouse, 1
		ate	1	now	[1]	now, 1
		the	1	quick	[1]	quick, 1
		mouse	1	the	[1,1,1]	the, 3
file1	how now brown cow	how	1			
		now	1			
		brown	1			
		cow	1			

Introduction: Page Rank

Page Rank: PageRank is an algorithm used to measure the importance of web pages in search engine ranking. It assigns a numerical weight to each page based on the number and quality of incoming links. By utilizing Spark's distributed computing capabilities, the PageRank algorithm can be applied to a large graph of web pages to determine the relative importance of each page.



Introduction: Page Rank Cntd.

The Process of Calculating PageRank :

Initialize each page's rank to 1.0: At the beginning of the algorithm, assign an initial rank of 1.0 to every page in the web graph.

Iterative contribution calculation :

On each iteration, each page (p) sends a contribution of rank(p) divided by the number of neighbors (pages it has links to) to its neighbors. The contribution represents the proportion of the current page's rank that it distributes to its neighboring pages.

Update each page's rank :

After receiving contributions from its neighbors, each page's rank is updated using the formula: rank = 0.15 + 0.85 * contributionsReceived.

Implementation

Authenticate with Google Cloud Platform (GCP)

```
$ gcloud auth login
```

Add the Helm stable repository:

```
$ helm repo add stable https://charts.helm.sh/stable
```

Install the NFS server provisioner using Helm:

```
$ helm install nfs stable/nfs-server-provisioner \set persistence.enabled=true,persistence.size=5Gi
```

Implementation Cntd.

Create a spark-pvc.yaml file. (Added in this repository)

```
$ vim spark-pvc.yaml
```

Apply the PersistentVolumeClaim (PVC) using the following command:

```
$ kubectl apply -f spark-pvc.yaml
```

Copy the Spark examples JAR file to a local directory using Docker:

```
$ docker run -v /tmp:/tmp -it bitnami/spark -- find /opt/bitnami/spark/examples/jars/ -name spark-examples* -exec cp {} /tmp/my.jar \;
```

Implementation Cntd.

Create a test file with a sample text:

```
$ echo "how much wood could a woodpecker chuck if a woodpecker could chuck wood" > /tmp/test.txt
```

Copy the JAR file and test file to the Spark data pod:

```
$ kubectl cp /tmp/my.jar spark-data-pod:/data/my.jar
```

```
$ kubectl cp /tmp/test.txt spark-data-pod:/data/test.txt
```

Create a spark-chart.yaml file(Added in this repository)

```
$ vim-spark-chart.yaml
```

Implementation Cntd.

Add the Bitnami Helm repository:

```
$ helm repo add bitnami https://charts.bitnami.com/bitnami
```

Install Spark using Helm with the provided configuration:

```
$ helm install spark bitnami/spark -f spark-chart.yaml
```

Get the external IP address of the Spark service:

```
$ kubectl get svc -l "app.kubernetes.io/instance=spark,app.kubernetes.io/name=spark"
```

Implementation: Finding Word Count

Run the Spark client using kubectl to submit a word count job:

```
$ kubectl run --namespace default spark-client --rm --tty -i --restart='Never' \
  --image docker.io/bitnami/spark:3.4.1-debian-11-r3 \
  -- spark-submit --master spark://<SPARK_MASTER_IP>:7077 \
  --deploy-mode cluster \
  --class org.apache.spark.examples.JavaWordCount \
  /data/my.jar /data/test.txt
```

Implementation: Finding Word Count Cntd.

Get the name of the Spark worker pod:

```
$ kubectl get pods -o wide | grep <SPARK_WORKER_IP>
```

Replace <SPARK_WORKER_IP> with the IP address obtained in previous step.

Enter the Spark worker pod:

```
$ kubectl exec -it spark-worker-1 -- bash
```

Navigate to the Spark work directory:

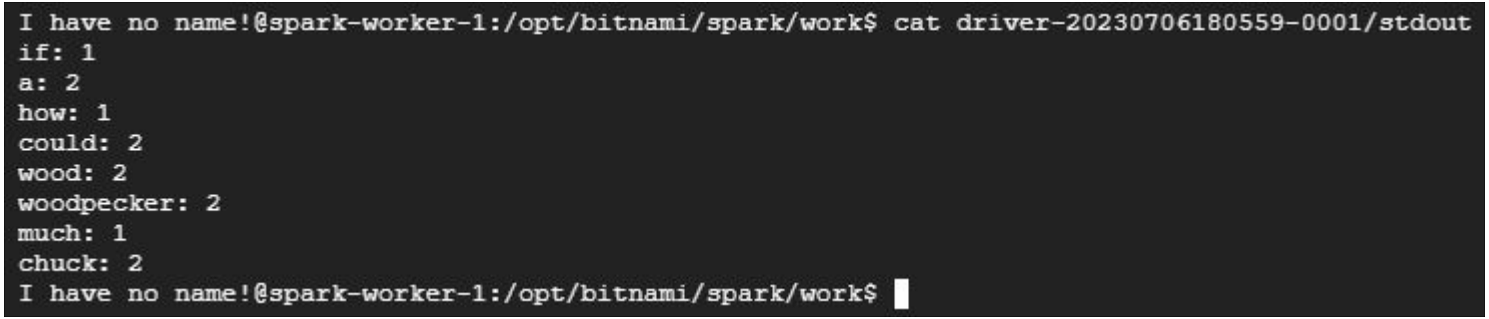
```
I have no name!@spark-worker-1:/opt/bitnami/spark

$ cd /opt/bitnami/spark/work
```

Output: Finding Word Count Cntd.

View the output file generated by the word count job:

cat <Submission ID >/stdout

**<Submission_Id> is the driver id the completed driver you can get from external IP.



```
I have no name!@spark-worker-1:/opt/bitnami/spark/work$ cat driver-20230706180559-0001/stdout
if: 1
a: 2
how: 1
could: 2
wood: 2
woodpecker: 2
much: 1
chuck: 2
I have no name!@spark-worker-1:/opt/bitnami/spark/work$
```

Implementation: Page Rank

Access the Spark master pod:

$ kubectl exec -it spark-master-0 -- bash

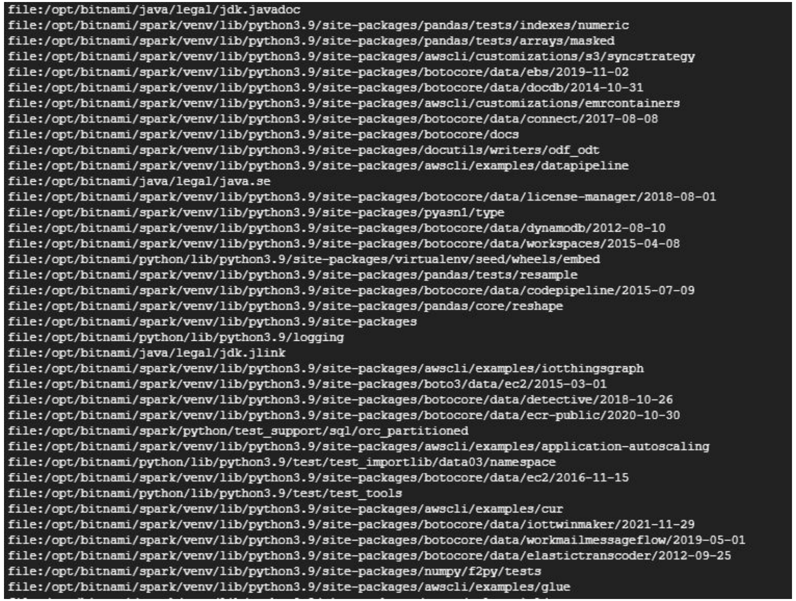
Navigate to the Python examples directory:

I have no name!@spark-master-0:/opt/bitnami/spark$

cd /opt/bitnami/spark/examples/src/main/python

Submit the PageRank job using spark-submit:

spark-submit pagerank.py /opt 2



```
file:/opt/bitnami/java/legal/jdk.javadoc
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/pandas/tests/indexes/numeric
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/pandas/tests/arrays/masked
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/awscli/customizations/s3/syncstrategy
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/botocore/data/ebs/2019-11-02
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/botocore/data/docdb/2014-10-31
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/awscli/customizations/emrcontainers
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/botocore/data/connect/2017-08-08
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/botocore/docs
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/docutils/writers/odf_odt
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/awscli/examples/datapipeline
file:/opt/bitnami/java/legal/java.se
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/botocore/data/license-manager/2018-08-01
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/pyasn1/type
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/botocore/data/dynamodb/2012-08-10
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/botocore/data/workspaces/2015-04-08
file:/opt/bitnami/python/lib/python3.9/site-packages/virtualenv/seed/wheels/embed
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/pandas/tests/resample
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/botocore/data/codepipeline/2015-07-09
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/pandas/core/reshape
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages
file:/opt/bitnami/python/lib/python3.9/logging
file:/opt/bitnami/java/legal/jdk.jlink
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/awscli/examples/iotthingsgraph
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/boto3/data/ec2/2015-03-01
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/botocore/data/detective/2018-10-26
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/botocore/data/ecr-public/2020-10-30
file:/opt/bitnami/spark/python/test_support/sql/orc_partitioned
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/awscli/examples/application-autoscaling
file:/opt/bitnami/python/lib/python3.9/test/test_importlib/data03/namespace
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/botocore/data/ec2/2016-11-15
file:/opt/bitnami/python/lib/python3.9/test/test_tools
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/awscli/examples/cur
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/botocore/data/iottwinmaker/2021-11-29
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/botocore/data/workmailmessageflow/2019-05-01
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/botocore/data/elastictranscoder/2012-09-25
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/numpy/f2py/tests
file:/opt/bitnami/spark/venv/lib/python3.9/site-packages/awscli/examples/glue
```