# Final Project

Wei Wang

## Mount data from s3 bucket

1. *Download data*
2. *Upload to s3 bucket (AWS CLI would be fast  and stable)*
3. *Mount data in databricks*

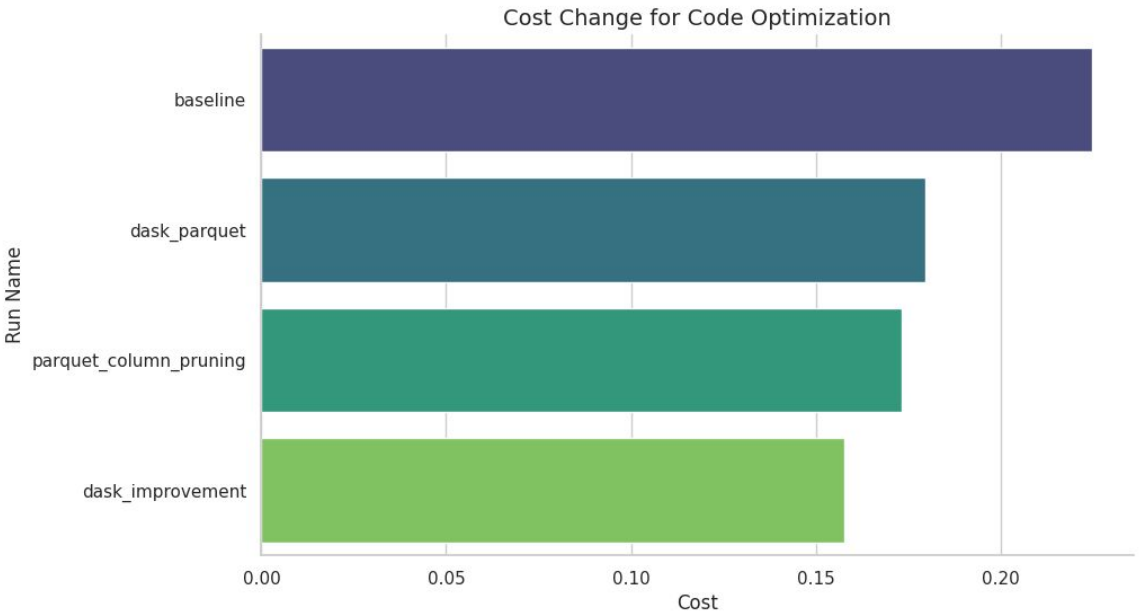
## Create jobs to do experiments

1. *Code optimization*
2. *Different languages*
3. *Different clusters*

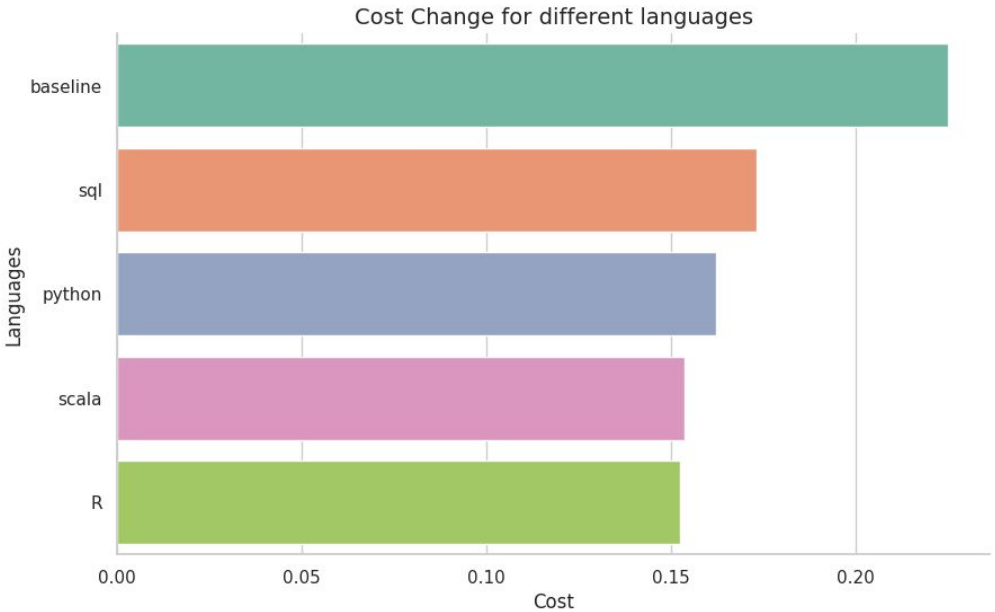
# Jobs

| Name | Tags | Created by | Trigger | Recent runs |
|---|---|---|---|---|
| baseline | | wei wang | | |
| baseline_job | | wei wang | | |
| baseline_job_general | | wei wang | | |
| dask_improvement | | wei wang | | |
| dask_improvement_general | | wei wang | | |
| dask_parquet | | wei wang | | |
| dask_parquet_general | | wei wang | | |
| parquet_column_pruning | | wei wang | | |
| parquet_column_pruning_gengeral | | wei wang | | |
| R | | wei wang | | |
| R_general | | wei wang | | |
| scala | | wei wang | | |
| scala_general | | wei wang | | |
| sql | | wei wang | | |
| sql_general | | wei wang | | |

# Code optimization



# Languages


Cost Change for different languages
baseline
sql
python
scala
R
Languages
0.00
0.05
0.10
0.15
0.20
Cost

# Clusters

1. *All-purpose cluster*

2. *Job cluster*

**Performance**

Databricks Runtime Version

13.3 LTS (includes Apache Spark 3.4.1, Scala 2.12)

☑ Use Photon Acceleration

| Worker type | | Min workers | Max workers | Current |
|---|---|---|---|---|
| i3.xlarge | 30.5 GB Memory, 4 Cores | 2 | 8 | 2 |

New Use fleet instance types for improved spot placement and availability Learn more

Driver type

i3.xlarge 30.5 GB Memory, 4 Cores

**Performance**

**Databricks runtime version**

Runtime: 13.3 LTS (Scala 2.12, Spark 3.4.1)

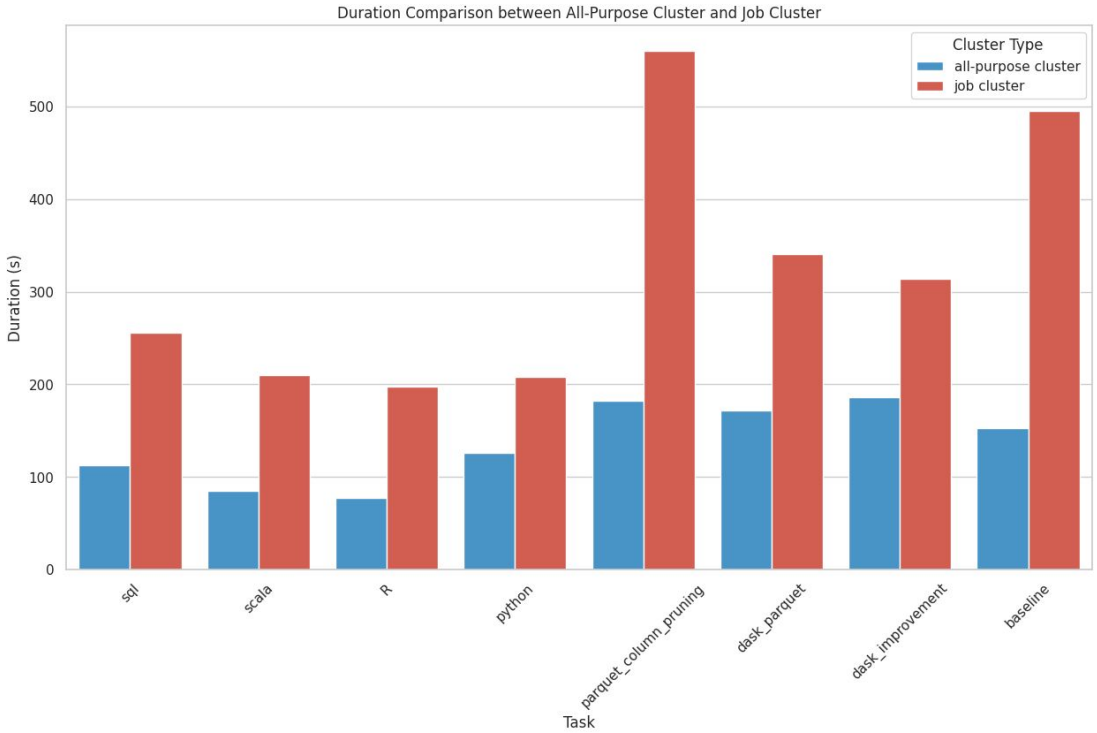
☑ Use Photon Acceleration

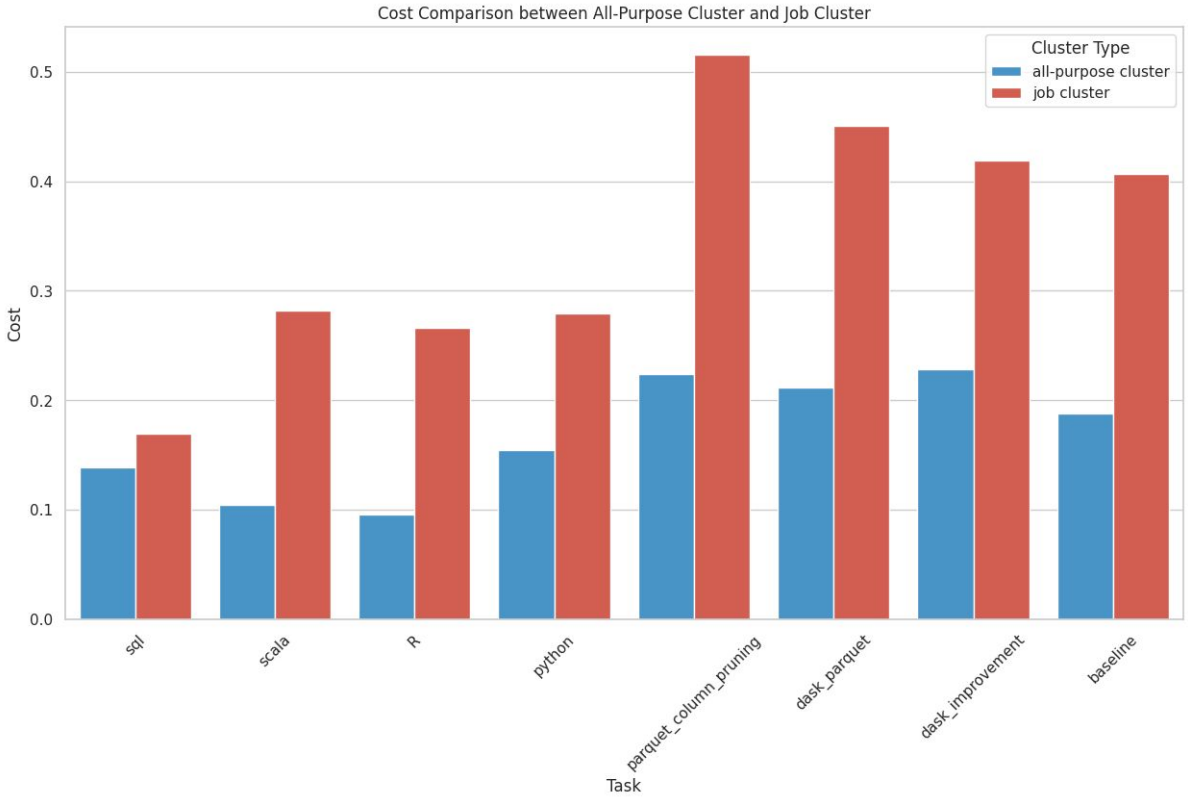
| Worker type | | Workers |
|---|---|---|
| i3.xlarge | 30.5 GB Memory, 4 Cores | 8 |

New Use fleet instance types for improved spot placement and availability Learn more

Driver type

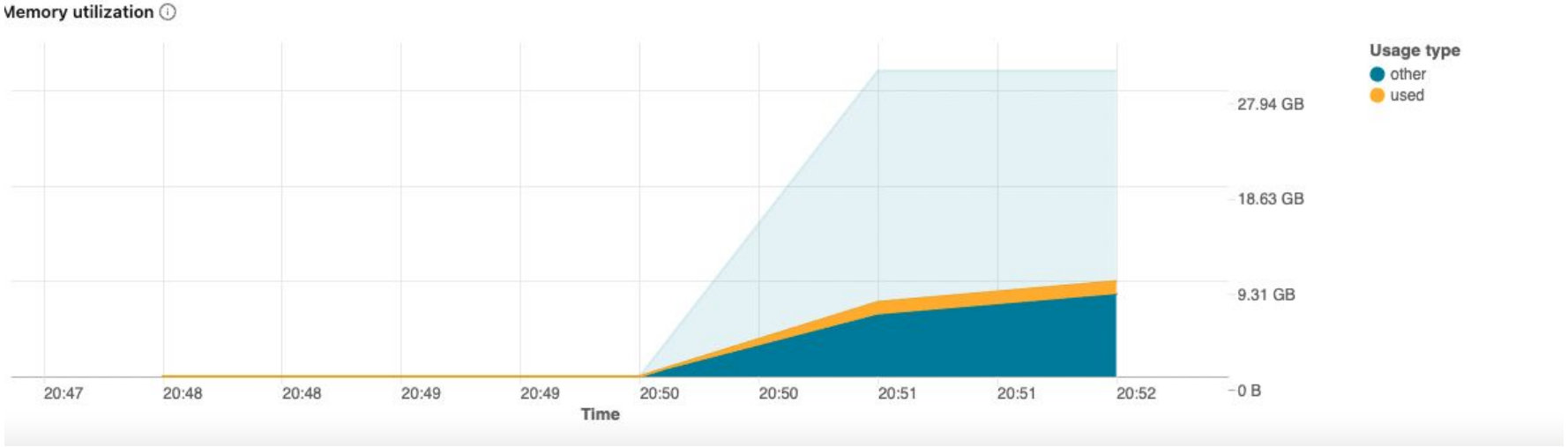
Same as worker 30.5 GB Memory, 4 Cores


Duration Comparison between All-Purpose Cluster and Job Cluster
Cluster Type
all-purpose cluster
job cluster
Duration (s)
0
100
200
300
400
500
sql
scala
R
python
parquet_column_pruning
dask_parquet
dask_improvement
baseline
Task


Cost Comparison between All-Purpose Cluster and Job Cluster
Cluster Type
all-purpose cluster
job cluster
Cost
0.0
0.1
0.2
0.3
0.4
0.5
sql
scala
R
python
parquet_column_pruning
dask_parquet
dask_improvement
baseline
Task

| | sku_name | avg(usage_quantity) | Cost |
|---|---|---|---|
| 1 | PREMIUM_JOBS_COMPUTE_(PHOTON) | 8.4561889444444450000000 | 1.268428 |
| 2 | PREMIUM_ALL_PURPOSE_COMPUTE_(PHOTON) | 3.3428450000000000000000 | 1.838565 |

# Job Cluster


Memory utilization
Usage type
other
used
27.94 GB
18.63 GB
9.31 GB
0 B
20:47
20:48
20:48
20:49
20:49
20:50
20:50
20:51
20:51
20:52
Time

# Different clusters

Workflows & Streaming

Jobs

Starting at $0.07 / DBU

Run data engineering pipelines to build data lakes and manage data at scale

All Purpose Compute for Interactive Workloads

Starting at $0.40 / DBU

1. *All-purpose cluster: Quick and Convenient, Higher Cost*

2. *Job cluster: Economical and On-Demand, Initial Latency*