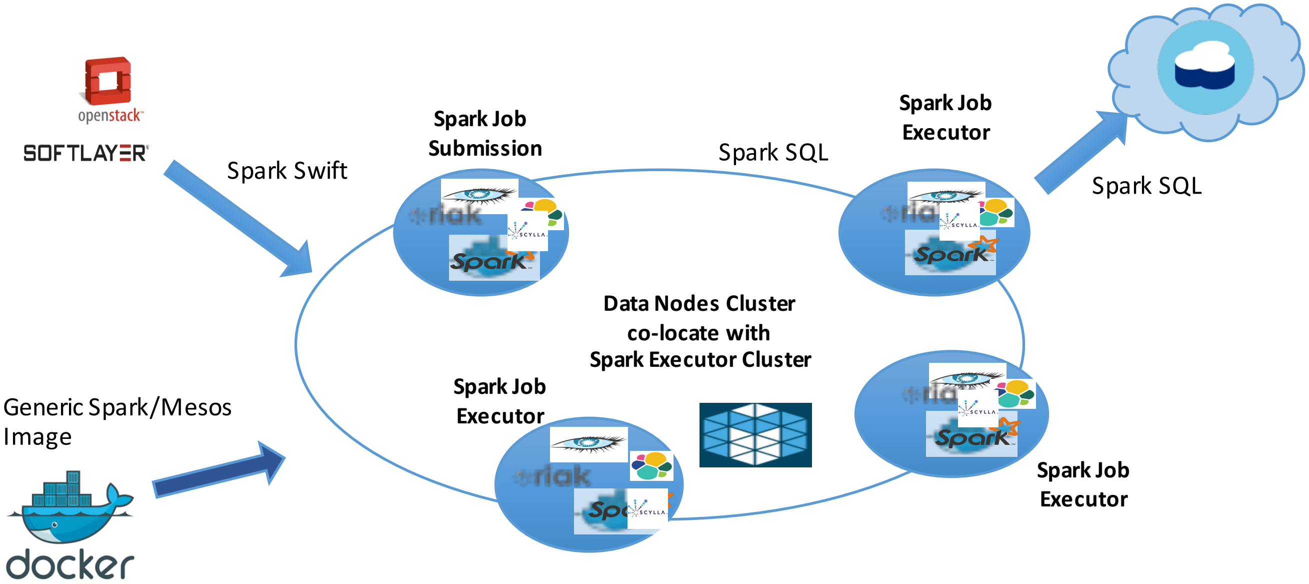
Spark SQL Workload on Mesos

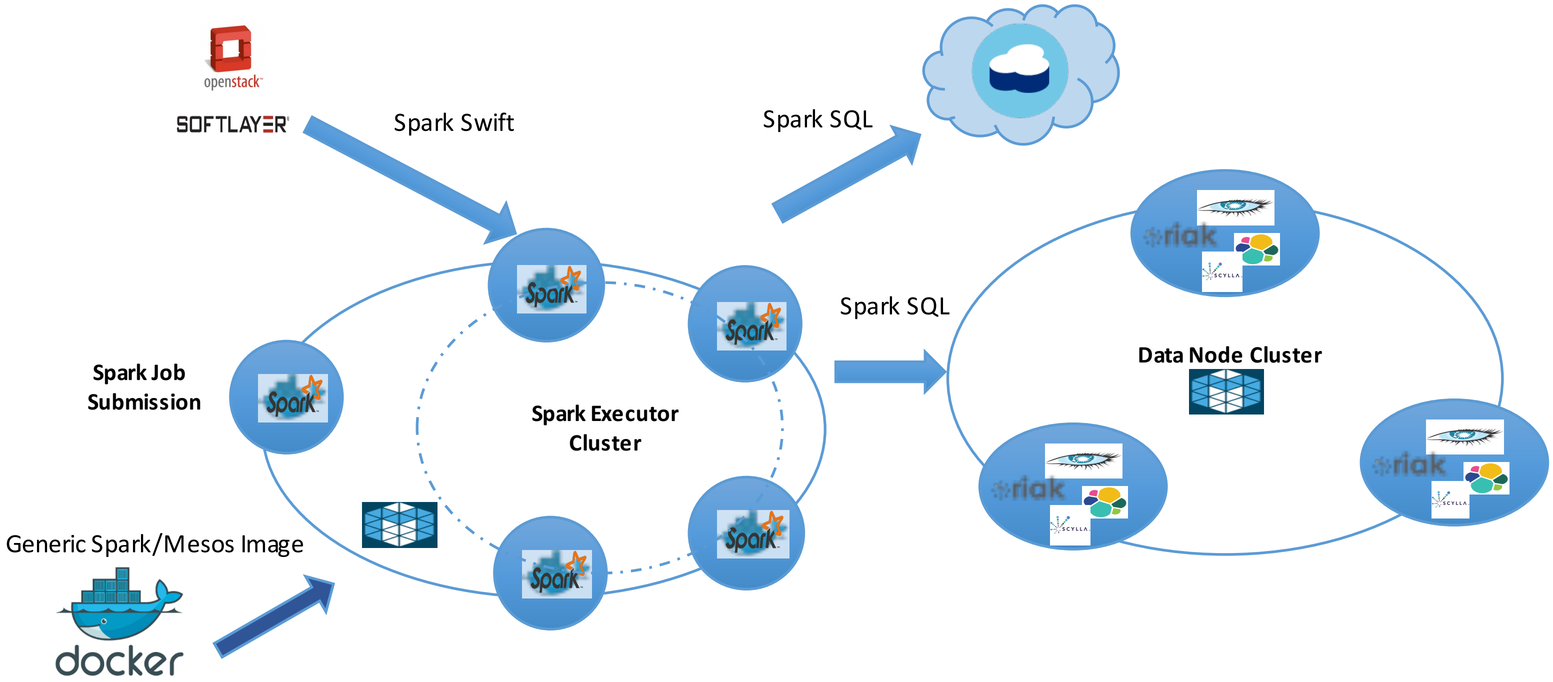
Yang Lei (yanglei@us.ibm.com)

Nov. 9th, 2016

A Sample Topology of DataStore Benchmark(Spark SQL): Shared Cluster

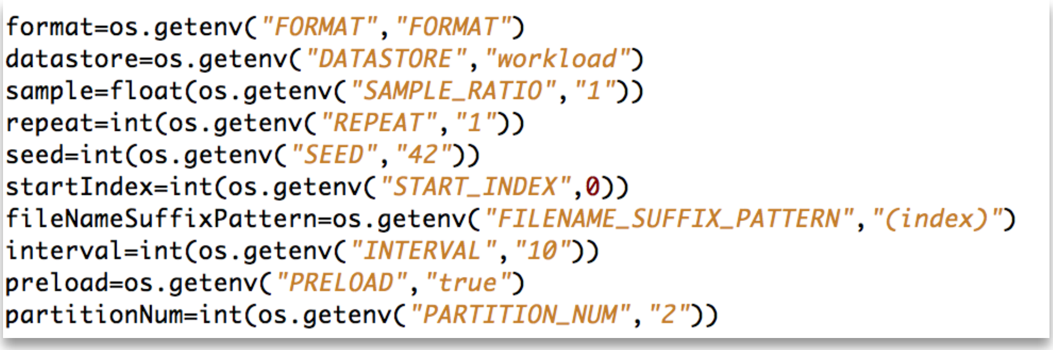


A Topology Variation of DataStore Benchmark(Spark SQL): Dedicated Data Cluster

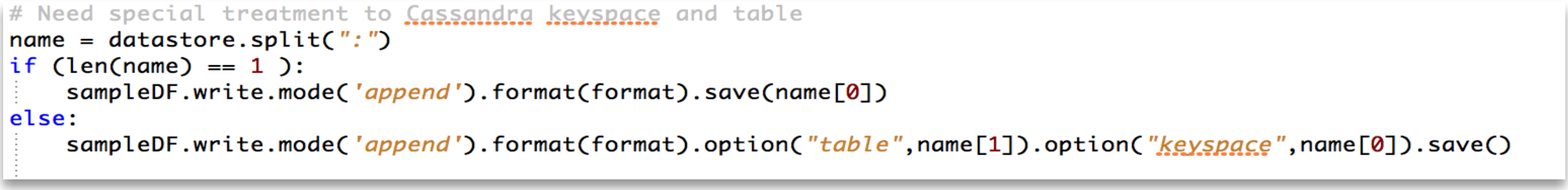


Benchmark Details: Unified code with Abstractions

- DataStore abstraction into **format** (Spark SQL) and **name** (Table, Database, KeySpace…)
- **Sample ratio**, **Seed** and **Repeat** are defined to control the size of the data for each write.
- **Interval** is defined for wait time between each write.
- **Partition** can be defined to force dataframe repartition



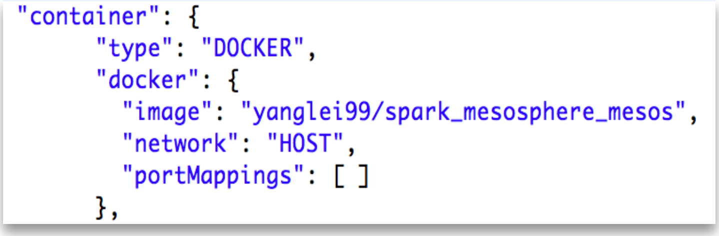
```
format=os.getenv("FORMAT","FORMAT")
datastore=os.getenv("DATASTORE","workload")
sample=float(os.getenv("SAMPLE_RATIO","1"))
repeat=int(os.getenv("REPEAT","1"))
seed=int(os.getenv("SEED","42"))
startIndex=int(os.getenv("START_INDEX",0))
fileNameSuffixPattern=os.getenv("FILENAME_SUFFIX_PATTERN","(index)")
interval=int(os.getenv("INTERVAL","10"))
preload=os.getenv("PRELOAD","true")
partitionNum=int(os.getenv("PARTITION_NUM","2"))
```



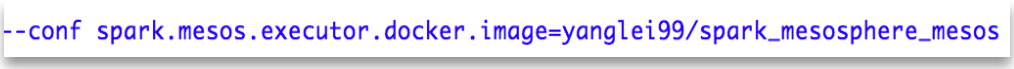
```
# Need special treatment to Cassandra keyspace and table
name = datastore.split(":")
if (len(name) == 1 ):
    sampleDF.write.mode('append').format(format).save(name[0])
else:
    sampleDF.write.mode('append').format(format).option("table",name[1]).option("keyspace",name[0]).save()
```

Benchmark Details: Simple & Scale with Customization

- One Docker image for both Spark job submission and Spark executor

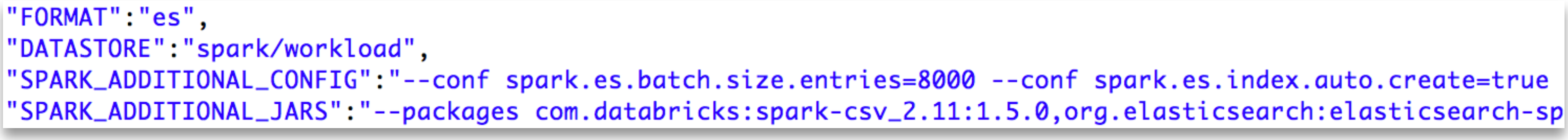


```
"container": {
      "type": "DOCKER",
      "docker": {
        "image": "yanglei99/spark_mesosphere_mesos",
        "network": "HOST",
        "portMappings": [ ]
      },
```



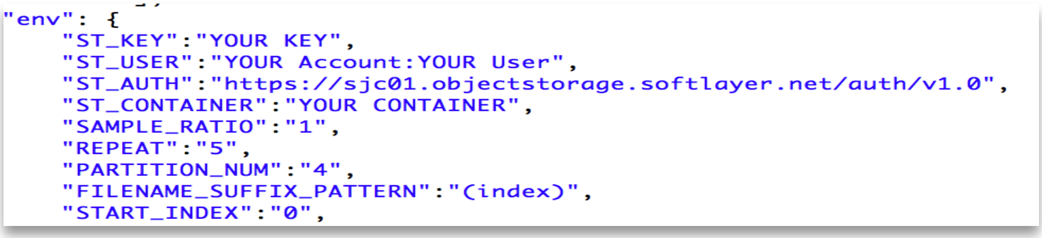
```
--conf spark.mesos.executor.docker.image=yanglei99/spark_mesosphere_mesos
```

- DataStore specific settings, e.g. hosts, credentials, dependent library ..., defined in Marathon JSON



```
"FORMAT":"es",
"DATASTORE":"spark/workload",
"SPARK_ADDITIONAL_CONFIG":"--conf spark.es.batch.size.entries=8000 --conf spark.es.index.auto.create=true
"SPARK_ADDITIONAL_JARS":"--packages com.databricks:spark-csv_2.11:1.5.0,org.elasticsearch:elasticsearch-sp
```

- Run variance can be defined through environment variable



```
"env": {
    "ST_KEY":"YOUR KEY",
    "ST_USER":"YOUR Account:YOUR User",
    "ST_AUTH":"https://sjc01.objectstorage.softlayer.net/auth/v1.0",
    "ST_CONTAINER":"YOUR CONTAINER",
    "SAMPLE_RATIO":"1",
    "REPEAT":"5",
    "PARTITION_NUM":"4",
    "FILENAME_SUFFIX_PATTERN":"(index)",
    "START_INDEX":"0",
```