Untitled2

July 14, 2022

[1]:
```
from pyspark import pandas as ps
import re
import numpy as np
import os
import sys
#import pandas as pd
import multiprocessing

from pyspark import SparkContext, SparkConf
from pyspark.sql import SparkSession
from pyspark.sql.functions import concat, concat_ws, lit, col, trim, expr
from pyspark.sql.types import StructType, StructField, StringType,IntegerType

os.environ["PYARROW_IGNORE_TIMEZONE"]="1"

number_cores = int(multiprocessing.cpu_count())

mem_bytes = os.sysconf('SC_PAGE_SIZE') * os.sysconf('SC_PHYS_PAGES')  # e.g.␣
↪4015976448
memory_gb = int(mem_bytes/(1024.**3))  # e.g. 3.74


def get_spark_session(app_name: str, conf: SparkConf):
    conf.setMaster('local[{}]'.format(number_cores))
    conf \
      .set('spark.driver.memory', '{}g'.format(memory_gb)) \
      .set("spark.sql.repl.eagerEval.enabled", "True") \
      .set("spark.sql.adaptive.enabled", "True") \
      .set("spark.serializer", "org.apache.spark.serializer.KryoSerializer") \
      .set("spark.sql.repl.eagerEval.maxNumRows", "10000")\

    return SparkSession.builder.appName(app_name).config(conf=conf).
↪getOrCreate()

spark = get_spark_session("Sid", SparkConf())
```

```
WARNING:root:'PYARROW_IGNORE_TIMEZONE' environment variable was not set. It is
required to set this environment variable to '1' in both driver and executor
```

```
sides if you use pyarrow>=2.0.0. pandas-on-Spark will set it for you but it does
not work if there is a Spark context already launched.
```

[2]:
```
spark
```

[2]:
```
<pyspark.sql.session.SparkSession at 0x7faab2fbf010>
```

[3]:
```
test = ps.read_csv("test.dat")
```

```
/opt/spark/python/pyspark/pandas/utils.py:976: PandasAPIOnSparkAdviceWarning: If
`index_col` is not specified for `read_csv`, the default index is attached which
can cause additional overhead.
  warnings.warn(message, PandasAPIOnSparkAdviceWarning)
```

[4]:
```
test
```

[4]:
```
                                    14/07/2022abc
0  PWJ   PWJABC 513213217ABC GM20 05.0000 6/20/39
1                                     #01000count
```

[5]:
```
from datetime import date

today = date.today()


day = today.strftime("%d/%m/%Y")
print(day)
```

```
14/07/2022
```

[6]:
```
testday = str(day)
```

[7]:
```
testday
```

[7]:
```
'14/07/2022'
```

[8]:
```
date_test = test.columns[0]
```

[9]:
```
df_date = date_test[:10]
```

[10]:
```
df_date
```

[10]:
```
'14/07/2022'
```

[11]:
```
day == df_date
```

[11]:
```
True
```

[12]:
```
test_str = test.loc[1][0]
```

[13]:
```
test_str
```

[13]:
```
'#01000count'
```

[14]:
```
test_str[2]
```

[14]:
```
'1'
```

[15]:
```
spark.stop()
```

[ ]:
```

```