What are PySpark Broadcast Variables and how are they used?

Authored by

stats writer

June 24, 2024

PySpark Broadcast Variables are a special type of read-only variable in the PySpark framework that allow for efficient sharing of data across multiple nodes in a cluster. These variables are distributed to every worker node and can be accessed locally, avoiding the need to transfer the data multiple times. This makes them particularly useful for improving the performance of operations such as joins and lookups. Broadcast variables are created by calling the broadcast() method on a regular Python variable and can be used in various transformations and actions in PySpark, such as map and filter functions. They can greatly enhance the speed and efficiency of data processing in distributed environments.

In PySpark RDD and DataFrame, Broadcast variables are read-only shared variables that are cached and available on all nodes in a cluster in-order to access or use by the tasks. Instead of sending this data along with every task, PySpark distributes broadcast variables to the workers using efficient broadcast algorithms to reduce communication costs.

Use case

Let me explain with an example when to use broadcast variables, assume you are getting a two-letter country state code in a file and you wanted to transform it to full state name, (for example CA to California, NY to New York e.t.c) by doing a lookup to reference mapping. In some instances, this data could be large and you may have many such lookups (like zip code e.t.c).

Instead of distributing this information along with each task over the network (overhead and time consuming), we can use the broadcast variable to cache this lookup info on each machine and tasks use this cached info while executing the transformations.

How does PySpark Broadcast work?

Broadcast variables are used in the same way for RDD, DataFrame.

When you run a PySpark RDD, DataFrame applications that have the Broadcast variables defined and used, PySpark does the following.

You should be creating and using broadcast variables for data that shared across multiple stages and tasks.

Note that broadcast variables are not sent to executors with `sc.broadcast(variable)` call instead, they will be sent to executors when they are first used.

How to create Broadcast variable

The PySpark Broadcast is created using the `broadcast(v)` method of the SparkContext class.

This method takes the argument v that you want to broadcast.

In PySpark shell

```
broadcastVar = sc.broadcast(Array(0, 1, 2, 3))
broadcastVar.value
```

PySpark RDD Broadcast variable example

Below is a very simple example of how to use broadcast variables on RDD. This example defines commonly used data (states) in a Map variable and distributes the variable using `SparkContext.broadcast()` and then use these variables on RDD map() transformation.

```
import pyspark
from pyspark.sql import SparkSession
```

spark = SparkSession.builder.appName('SparkByExamples.com').getOrCreate()

states = {"NY":"New York", "CA":"California", "FL":"Florida"}
broadcastStates = spark.sparkContext.broadcast(states)

data =

rdd = spark.sparkContext.parallelize(data)

def state_convert(code):
return broadcastStates.value

result = rdd.map(lambda x: (x,x,x,state_convert(x))).collect()
print(result)

Yields below output

```
[('James', 'Smith', 'USA', 'California'), ('Michael', 'Rose', 'USA', 'New York'), ('Robert', 'Williams', 'USA', 'California'), ('Maria', 'Jones', 'USA', 'Florida')]
```

PySpark DataFrame Broadcast variable example

Below is an example of how to use broadcast variables on DataFrame, similar to above RDD example, This also uses commonly used data (states) in a Map variable and distributes the variable using `SparkContext.broadcast()` and then use these variables on DataFrame map() transformation.

If you are not familiar with DataFrame, I will recommend to learn the DataFrame before proceeding further on this article.

```
import pyspark
from pyspark.sql import SparkSession
```

spark = SparkSession.builder.appName('SparkByExamples.com').getOrCreate()

states = {"NY":"New York", "CA":"California", "FL":"Florida"}
broadcastStates = spark.sparkContext.broadcast(states)

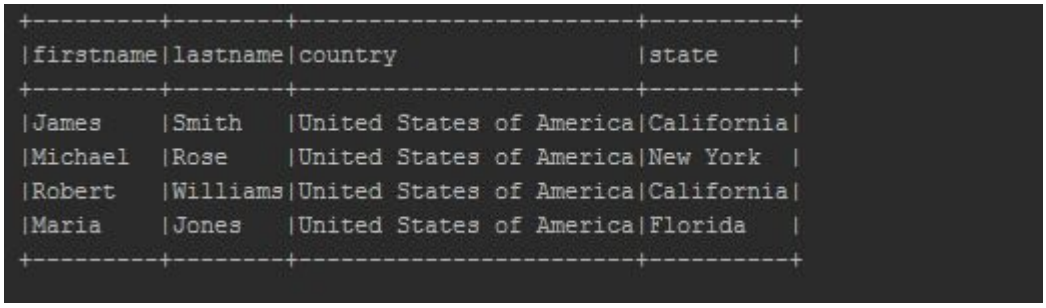
data =

columns =
df = spark.createDataFrame(data = data, schema = columns)
df.printSchema()
df.show(truncate=False)

def state_convert(code):
return broadcastStates.value

result = df.rdd.map(lambda x: (x,x,x,state_convert(x))).toDF(columns)
result.show(truncate=False)

Above example first creates a DataFrame, transform the data using broadcast variable and yields below output.



```
+---------+--------+-----------------------+----------+
|firstname|lastname|country                |state     |
+---------+--------+-----------------------+----------+
|James    |Smith   |United States of America|California|
|Michael  |Rose    |United States of America|New York  |
|Robert   |Williams|United States of America|California|
|Maria    |Jones   |United States of America|Florida   |
+---------+--------+-----------------------+----------+
```

You can also use the broadcast variable on the filter and joins. Below is a filter example.

```
# Broadcast variable on filter
filteDf= df.where((df.isin(broadcastStates.value)))
```

Conclusion

In this PySpark Broadcast variable article, you have learned what is Broadcast variable, it's advantage and how to use in RDD and Dataframe with Pyspark example.

Related Articles

Reference

Happy Learning !!