Làm việc với Spark Structure Streaming

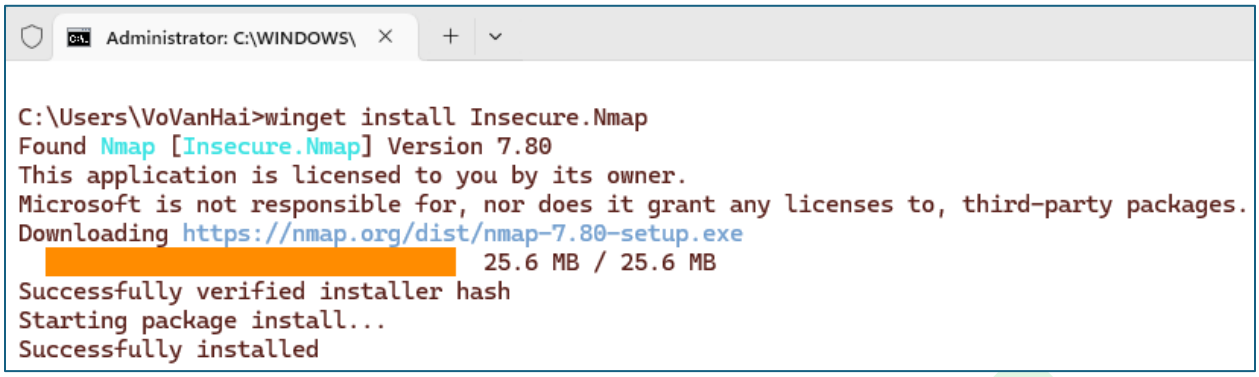
Bài 1

Đếm từ được gửi qua mạng sử dụng socket. Chương trình sẽ nhận dữ liệu dạng text được gửi qua socket tại cổng 9999. Khi có text gửi qua, ứng dụng sẽ đọc và thống kê sau đó in ra kết quả.

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import explode
from pyspark.sql.functions import split

if __name__ == '__main__':
    spark = SparkSession.builder \
        .appName("StructuredNetworkWordCount") \
        .getOrCreate()

    # Create a DataFrame representing the stream of input lines from the
connection to localhost:9999
    lines = spark.readStream \
        .format("socket") \
        .option("host", "localhost") \
        .option("port", 9999) \
        .load()

    # Split the lines into words
    words = lines.select(
        explode(
            split(lines.value, " ")
        ).alias("word")
    )

    # Generate running word count
    wordCounts = words.groupBy("word").count()

    # Start running the query that prints the running counts to the
console
    query = wordCounts \
        .writeStream \
        .outputMode("complete") \
        .format("console") \
        .start()
    query.awaitTermination()
```

Để thực thi được ứng dụng, ta cần 1 chương trình giả lập việc gửi dữ liệu vào một cổng xác định.

Chương trình đó là netcat. Bạn có thể cài trên Windows bằng lệnh winget như hình sau



Gõ nội dung sau câu lệnh này sau đó gõ enter sẽ send đoạn text vừa gõ lên port 9999

```
C:\Users\VoVanHai>ncat -lk 9999
this is sample lines
second line
another line
```

Thực thi chương trình

```
+-------+-----+
|   word|count|
+-------+-----+
|  lines|    1|
|     is|    1|
| second|    1|
| sample|    1|
|   this|    1|
|another|    1|
|   line|    2|
+-------+-----+
```

Bài 2

Giả sử chúng ta có thư mục *"/resources/csv_files/"* chứa các file csv với nội dung có 2 cột tên và lương. Dạng như sau

1	Donald OConnell	2600
2	Douglas Grant	2600
3	Jennifer Whalen	4400
4	Michael Hartstein	13000
5	Pat Fay	6000
6	Susan Mavris	6500

Viết một ứng dụng tính tổng lương của mỗi thành viên trong danh sách. Việc tính này luôn sẵn sàng. Mỗi khi có 1 file mới được copy vào thư mục, hệ thống sẽ tự động thực thi và tính cộng dồn.

Code hướng dẫn như sau

```python
from pyspark.sql import SparkSession
from pyspark.sql.types import StructType

if __name__ == '__main__':
    spark = SparkSession.builder.appName("sample").getOrCreate()
    # Define schema of the csv
    userSchema = (StructType()
                  .add("name", "string")
                  .add("salary", "integer"))

    # Spark streaming is waiting for csv files to be pushed to "/resources/csv_files/" folder.
    dfCSV = (spark.readStream
             .option(key: "sep", value: ",")
             .option(key: "header", value: "false")
             .schema(userSchema)
             .csv("resources/csv_files"))

    # We have defined the total salary per name. Note that this is a streaming DataFrame
    # which represents the running sum of the stream.

    dfCSV.createOrReplaceTempView("salary")
    totalSalary = spark.sql("select name,sum(salary) from salary group by name")

    # totalSalary = dfCSV.groupBy("name").sum("salary")

    # All that is left is to actually start receiving data and computing the counts.
    # To do this, we set it up to print the complete set of counts (specified by outputMode("complete"))
    # to the console every time they are updated. And then start the streaming computation using start().

    # Start running the query that prints the running counts to the console
    query = (totalSalary.writeStream
             .outputMode("complete")
             .format("console")
             .start())

    query.awaitTermination()
```

Kết quả dạng

```
-------------------------------------------
Batch: 0
-------------------------------------------
+-----------------+-----------+
|             name|sum(salary)|
+-----------------+-----------+
|      Steven King|      78445|
|Irene Mikkilineni|     108282|
|           Ki Gee|       2400|
|    Renske Ladwig|       3600|
|         John Seo|       2700|
```