Big Data Analytics

Dr Sirintra Vaiwsri | Email: sirintra.v@itm.kmutnb.ac.th

Streaming Processing

Streaming Processing (Chambers and Zaharia, 2018)

- Streaming processing continuously receives input and computes a result.
- It responds quickly and is more efficiency than batch processing.
- However, it is challenging if we need to process input that is not in sequence. For example, to process 2, 10, and then 5 where receiving 2, 5, and 10.

Streaming Processing (Antolínez García, 2023)

- Main characteristics of data streaming:
  - Continuous information
  - Unbounded information
  - High volume and velocity
  - Time sensitive
  - Heterogeneous data sources

Streaming Processing (Antolínez García, 2023)

- Streaming processing uses timestamps to order events in sequence.
  - Event time is based on the event that is generated by a device.
  - Ingestion time is based on the time of the event's arrival.
  - Processing time is based on the beginning time of the event process.

Spark Structured Streaming

Spark Structured Streaming (Antolínez García, 2023)

- Spark Structured Streaming provides a high-level manipulation.
- It also provides scalable and near-real-time streaming processing.
- It is built on top of the Spark SQL library and based on the Dataframe and Dataset APIs.

Spark Structured Streaming (Antolínez García, 2023)

- Concepts:
  - Input table: also called “unbounded input table”, means arrived data is appended as a new row in the table.
  - Result table: also called “unbounded output table”, means once new data input arrives, it is processed and the result table is updated.

Spark Structured Streaming (Antolínez García, 2023)

- Output modes:
  - Append mode - only new rows in the result table are written to the output sink.
  - Complete mode - write all rows in the result table every time that the data were processed.
  - Update mode - only write the updated rows to the output sink.

Programming Model for Structured Streaming

(Spark, 2023)

- New row(s) is appended to the input table (unbounded table) in every trigger interval, e.g. every 1 second.
- Data is then queried resulting in an updated result table.
- The result table can be then written out to the external storage as an output.

Dataframes Streaming API

Streaming Dataframes (Antolínez García, 2023)

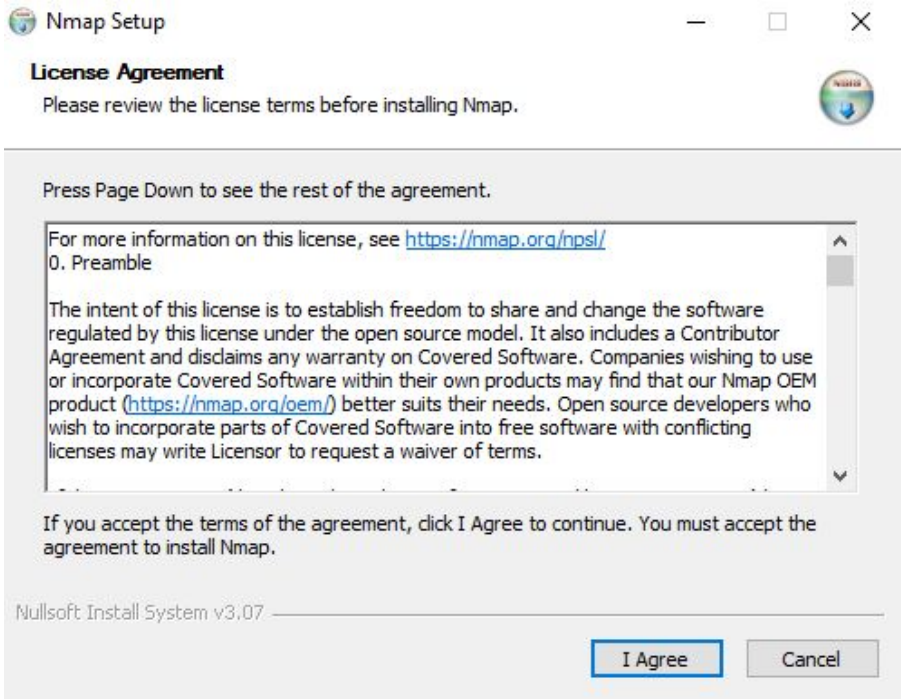
- Streaming Dataframes created using `SparkSession.readStream()`
- Input Sources:
  - Socket source
  - File source - a file such as CSV, JSON, or Parquet is read as a stream of data.
  - Kafka source - data is read from the Kafka source.

Socket Source

- Data can be ingested by listening to a socket connection.
- It is often used for testing.
- To execute, use NetCat which is in a Nmap package.
- Download Nmap (.exe) for Windows here.

Socket Source

- Install Nmap and Npcap
- We also install Npcap because the Nmap uses the Npcap library.



License Agreement

Please review the license terms before installing Nmap.

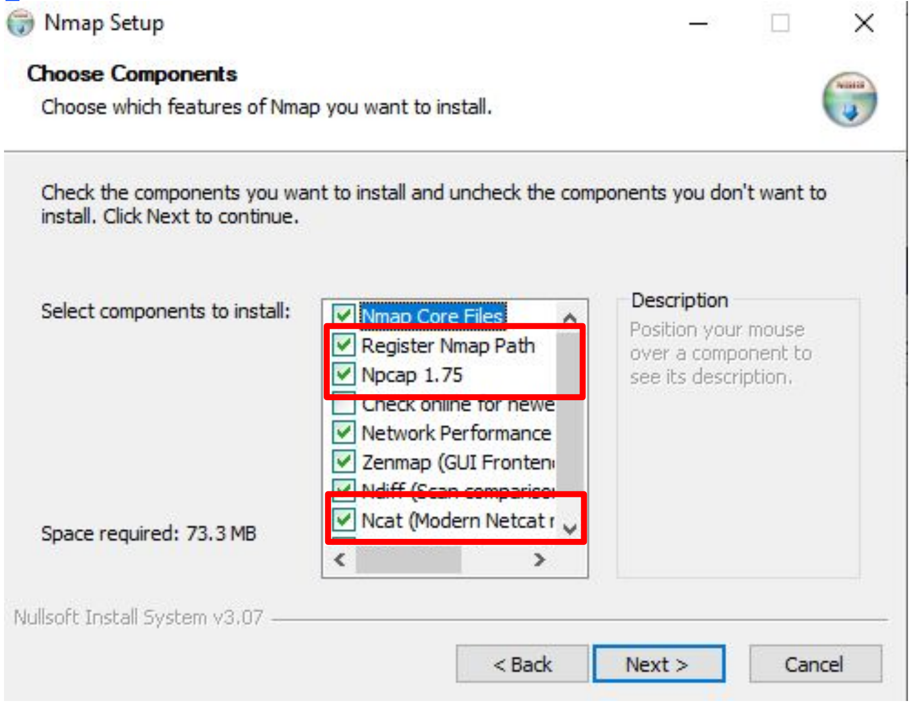
Press Page Down to see the rest of the agreement.

For more information on this license, see https://nmap.org/npsl/
0. Preamble

The intent of this license is to establish freedom to share and change the software regulated by this license under the open source model. It also includes a Contributor Agreement and disclaims any warranty on Covered Software. Companies wishing to use or incorporate Covered Software within their own products may find that our Nmap OEM product (https://nmap.org/oem/) better suits their needs. Open source developers who wish to incorporate parts of Covered Software into free software with conflicting licenses may write Licensor to request a waiver of terms.

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Nmap.

Socket Source



Choose Components

Choose which features of Nmap you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

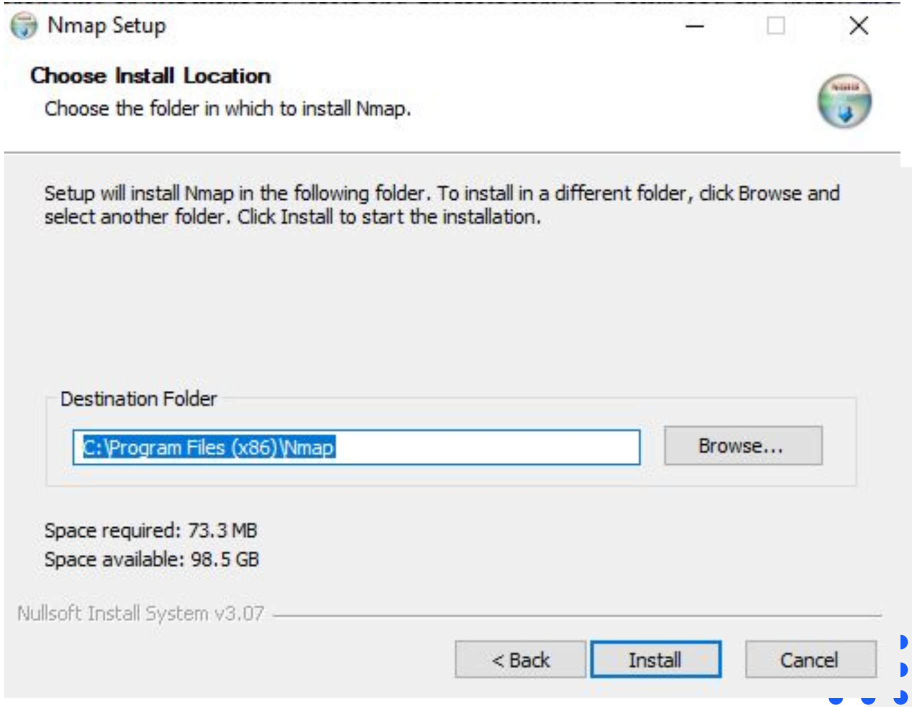
- ☑ Nmap Core Files
- ☑ Register Nmap Path
- ☑ Npcap 1.75
- □ Check online for newe
- ☑ Network Performance
- ☑ Zenmap (GUI Fronten
- ☑ Ndiff (Scan compariso
- ☑ Ncat (Modern Netcat r

Description

Position your mouse over a component to see its description.

Space required: 73.3 MB

Nullsoft Install System v3.07



Choose Install Location

Choose the folder in which to install Nmap.

Setup will install Nmap in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder

C:\Program Files (x86)\Nmap

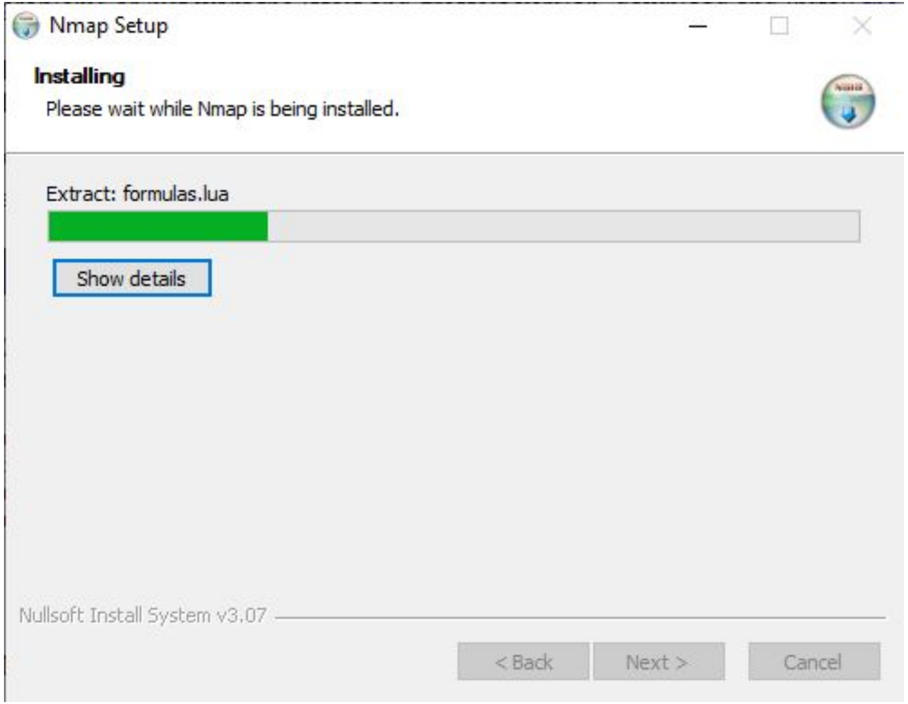
Space required: 73.3 MB

Space available: 98.5 GB

Nullsoft Install System v3.07

Dr Sirintra Vaiwsri

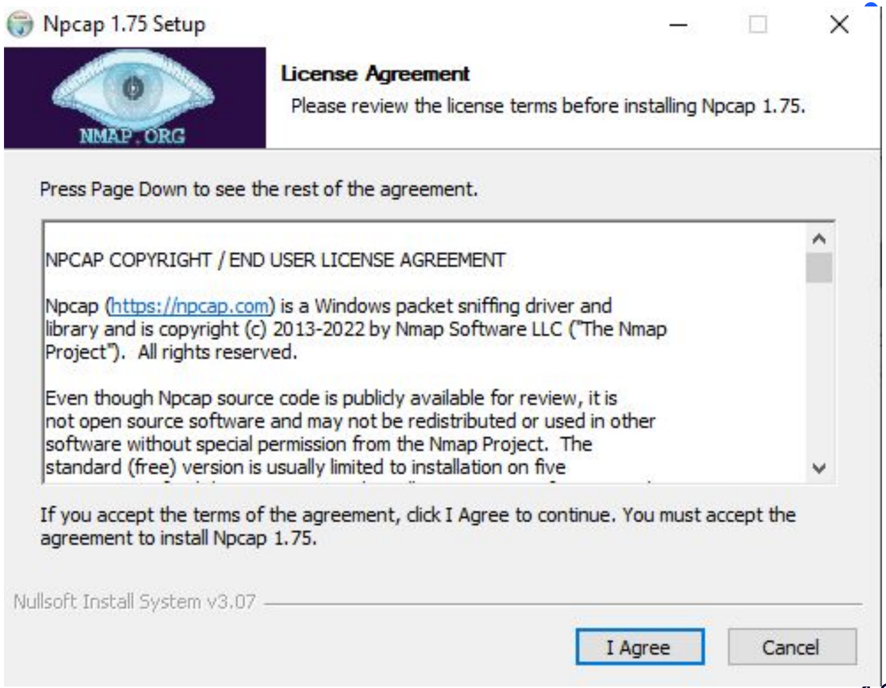
Socket Source



Installing

Please wait while Nmap is being installed.

Extract: formulas.lua



License Agreement

Please review the license terms before installing Npcap 1.75.

Press Page Down to see the rest of the agreement.

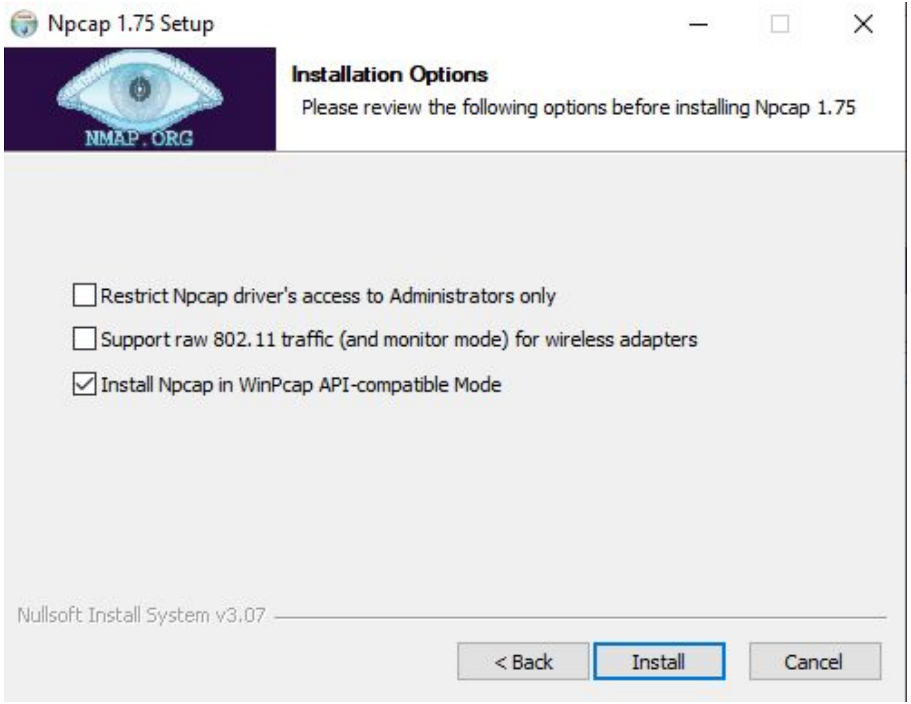
NPCAP COPYRIGHT / END USER LICENSE AGREEMENT

Npcap (https://npcap.com) is a Windows packet sniffing driver and library and is copyright (c) 2013-2022 by Nmap Software LLC ("The Nmap Project"). All rights reserved.

Even though Npcap source code is publicly available for review, it is not open source software and may not be redistributed or used in other software without special permission from the Nmap Project. The standard (free) version is usually limited to installation on five

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Npcap 1.75.

Socket Source

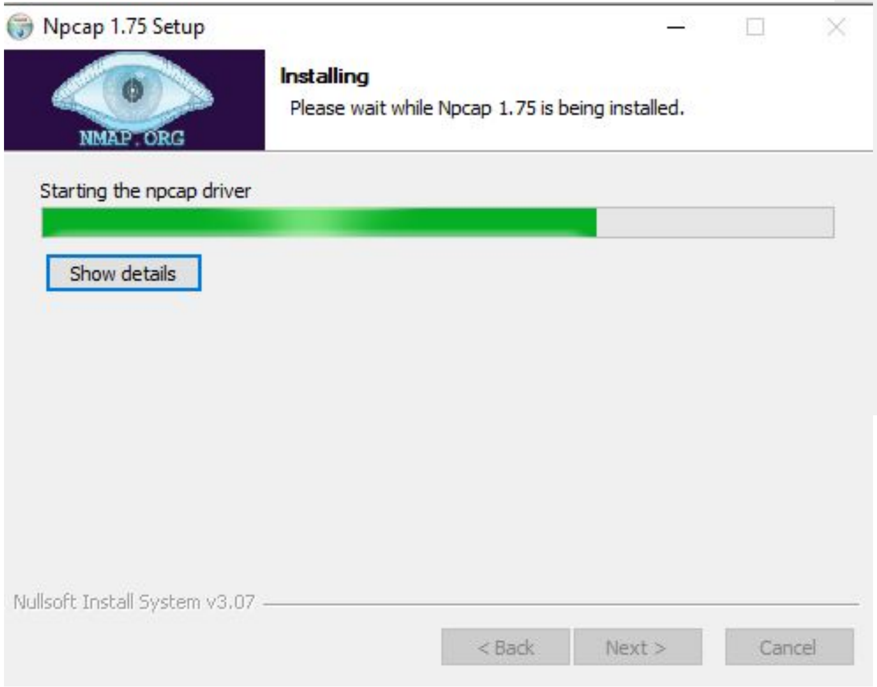


Npcap 1.75 Setup

Installation Options

Please review the following options before installing Npcap 1.75

- □ Restrict Npcap driver's access to Administrators only
- □ Support raw 802.11 traffic (and monitor mode) for wireless adapters
- ☑ Install Npcap in WinPcap API-compatible Mode



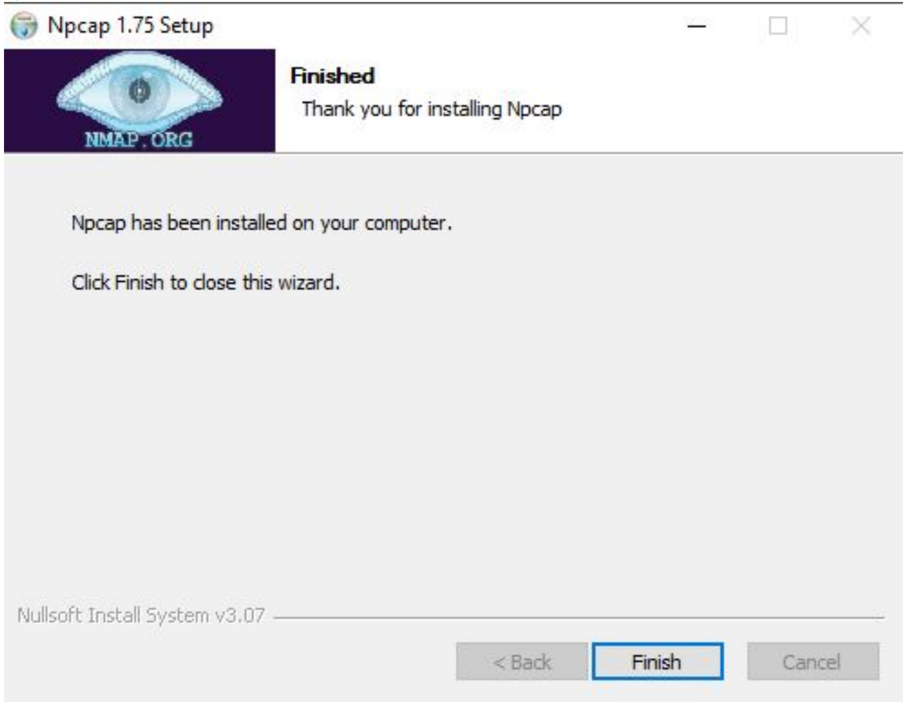
Npcap 1.75 Setup

Installing

Please wait while Npcap 1.75 is being installed.

Starting the npcap driver

Socket Source



Npcap 1.75 Setup

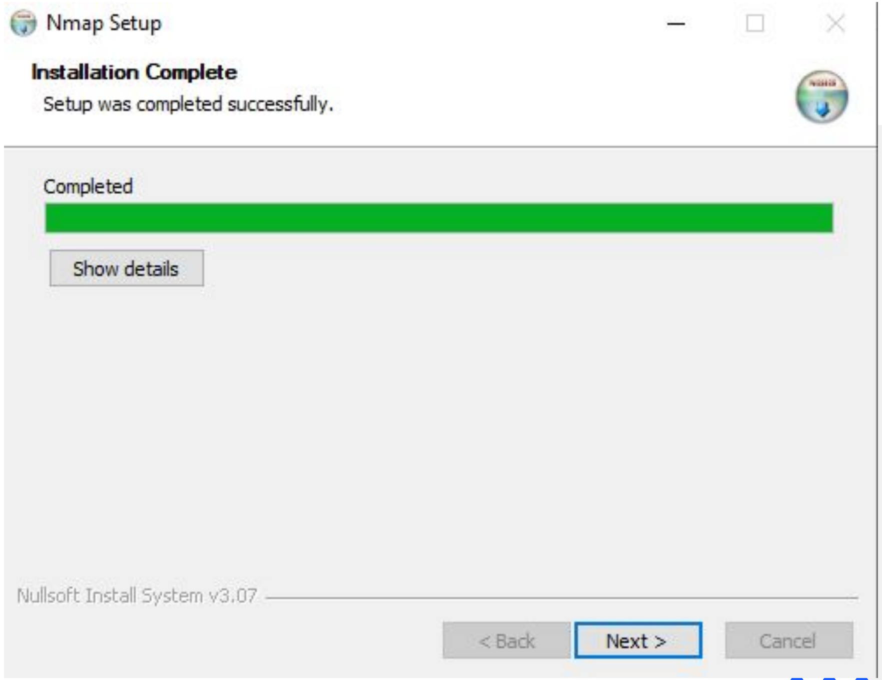
Finished

Thank you for installing Npcap

Npcap has been installed on your computer.

Click Finish to close this wizard.

Nullsoft Install System v3.07



Nmap Setup

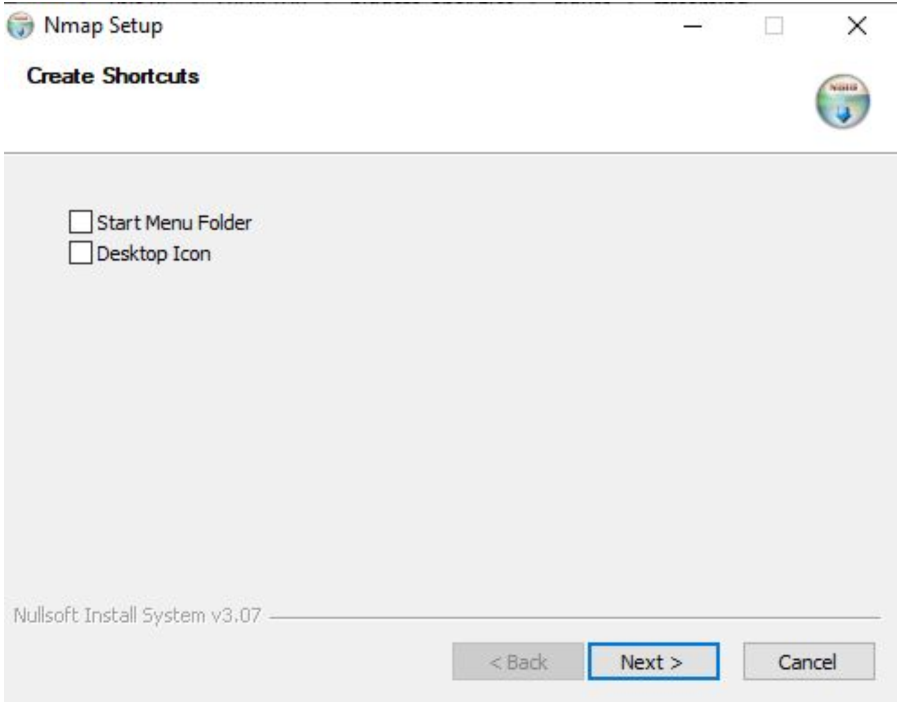
Installation Complete

Setup was completed successfully.

Completed

Nullsoft Install System v3.07

Socket Source



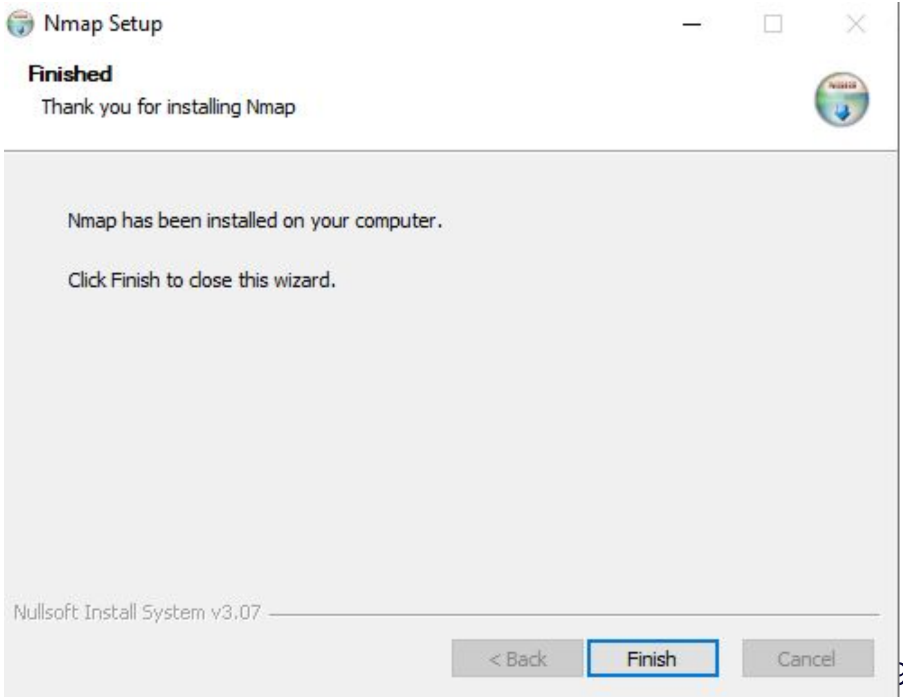
Nmap Setup

Create Shortcuts

□ Start Menu Folder

□ Desktop Icon

Nullsoft Install System v3.07



Nmap Setup

Finished

Thank you for installing Nmap

Nmap has been installed on your computer.

Click Finish to close this wizard.

Nullsoft Install System v3.07

Socket Source

Socket Source Implementation:

```
# Import Spark Session
# Import explode and split from the pyspark.sql.functions
# Create Spark Session
lines = <spark session>.readStream.format(“socket”).\
       option(“host”, “localhost”).\
       option(“port”, 9999).\
       load() #The function readStream() is used to return a
              # DataStreamReader for reading data streams as a
              # stream DataFrame.
```

Socket Source

```
words = lines.select(
                    explode(
                       split(lines.value, “ ”)
                    ).alias(“word”)
                 )
```

- The function explode() is used to return a new row for each value in a given array or map.
- The function split() is used to split value, where in this case we split value using a space (“ ”).
- The function alias() is used to name a new column.

Socket Source

```
wordCounts = words.groupBy(“word”).count() # Group
                                     # and count each word
query = wordCounts.writeStream.\
          outputMode(“<output mode>”).\
          format(“console”).start()
```

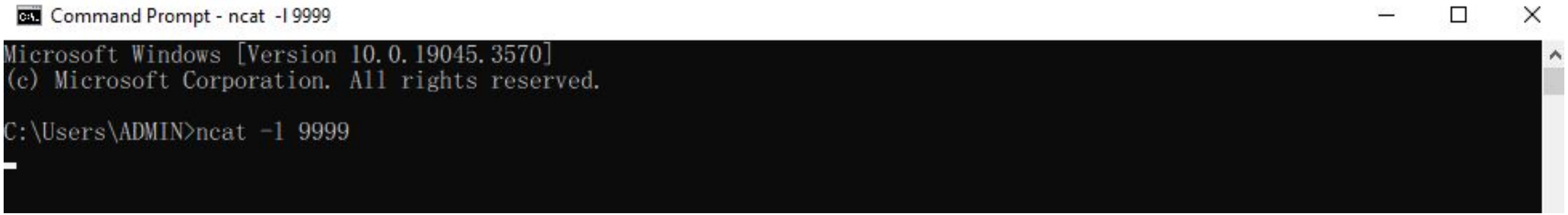
- The function writeStream() is used to save the streaming Dataframe out to the external storage, where in this case we use “console”.

```
query.awaitTermination()
```

- The function awaitTermination() is used to wait for the termination of the current query.

Socket Source

- Open the first command prompt
- Type the command ncat -l 9999 (-l is listen)

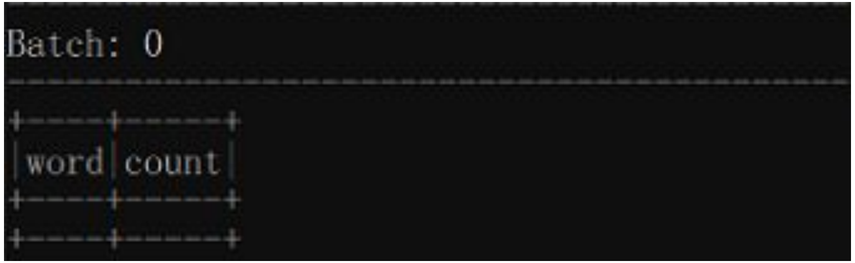


```
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ADMIN>ncat -l 9999
_
```

Dr Sirintra Vaiwsri

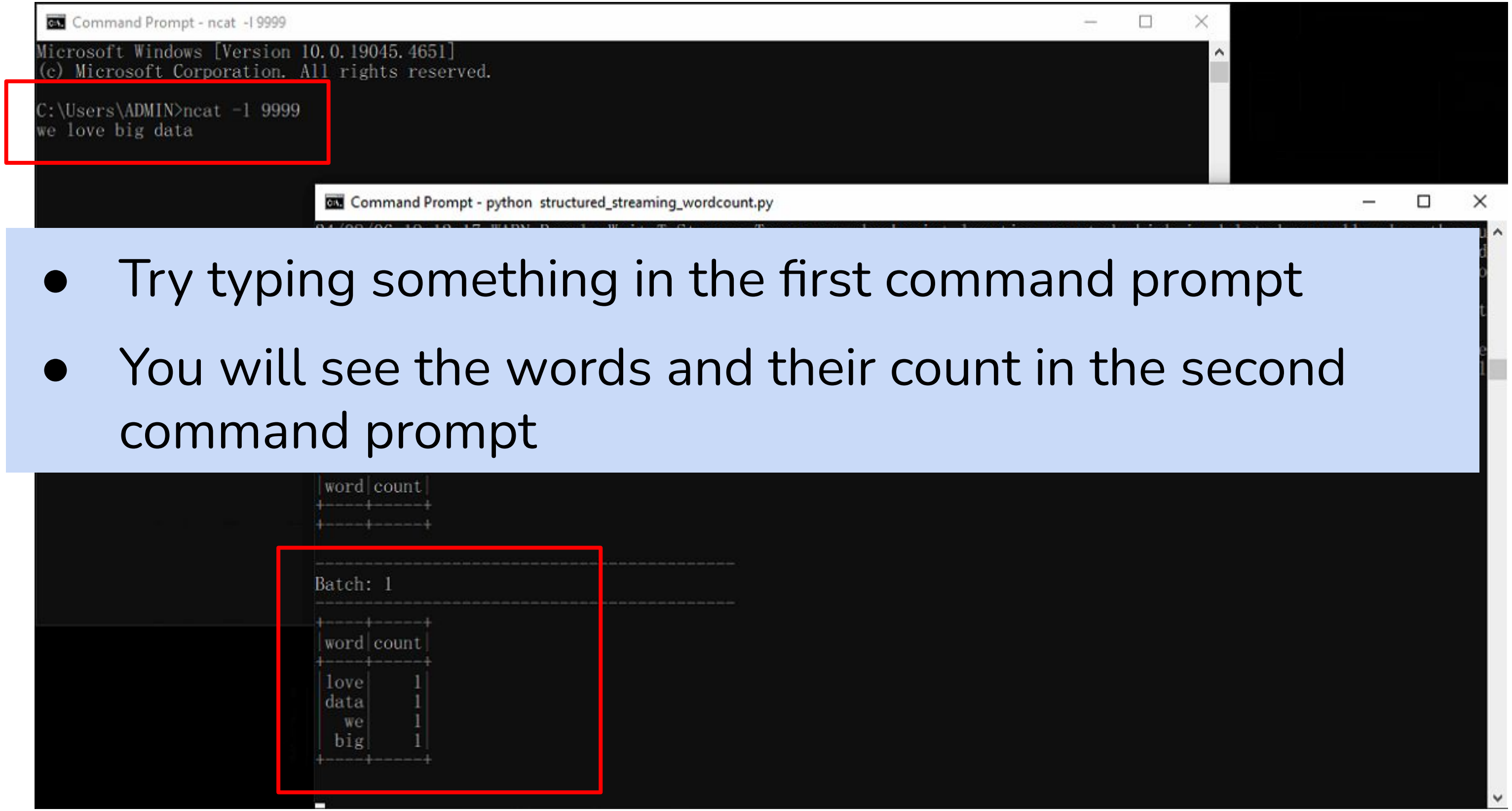
Socket Source

- Open the second command prompt
- Run the created Structured Streaming Python file
- First, you will see only an empty Dataframe.



```
Batch: 0
-------------------------------------------
+----+-----+
|word|count|
+----+-----+
+----+-----+
```

Socket Source

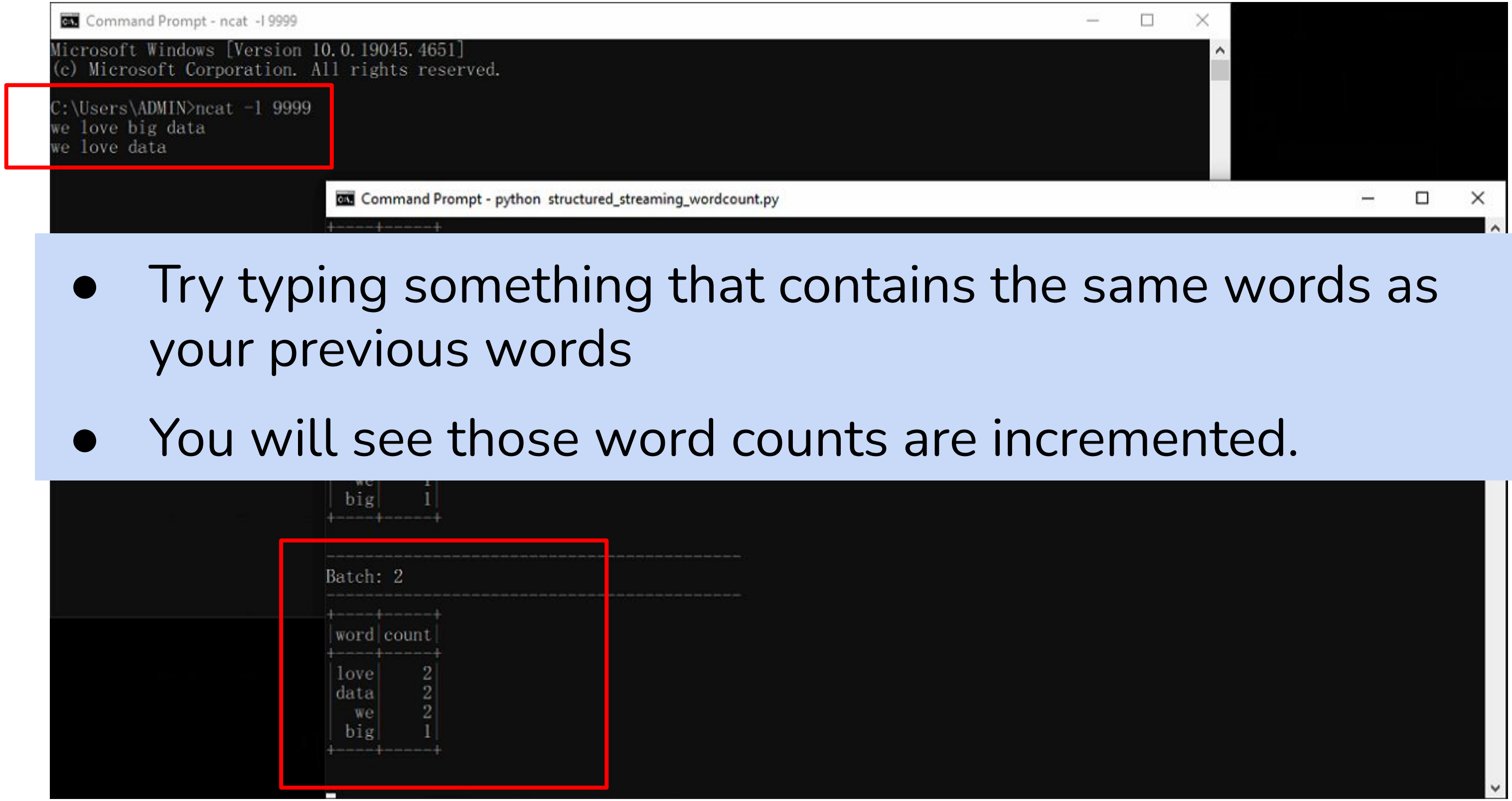

Command Prompt - ncat -l 9999
Command Prompt - python structured_streaming_wordcount.py

```
Microsoft Windows [Version 10.0.19045.4651]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ADMIN>ncat -l 9999
we love big data
```

- Try typing something in the first command prompt
- You will see the words and their count in the second command prompt

```
|word|count|
+----+-----+
+----+-----+

-------------------------------------------
Batch: 1
-------------------------------------------
+----+-----+
|word|count|
+----+-----+
|love|    1|
|data|    1|
|  we|    1|
| big|    1|
+----+-----+
```

Socket Source


Command Prompt - ncat -l 9999
Command Prompt - python structured_streaming_wordcount.py

```
Microsoft Windows [Version 10.0.19045.4651]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ADMIN>ncat -l 9999
we love big data
we love data
```

```
+----+-----+
```

```
|  big|    1|
+----+-----+

-------------------------------------------
Batch: 2
-------------------------------------------
+----+-----+
|word|count|
+----+-----+
|love|    2|
|data|    2|
|  we|    2|
| big|    1|
+----+-----+
```

- Try typing something that contains the same words as your previous words
- You will see those word counts are incremented.

File Source (Spark, 2023)

- A file is read as a stream of data.
- Once the file is modified, the file will be processed in the structured streaming
- Supported file formats are such as Text, CSV, JSON, etc.
- File source used in Structured Streaming requires a specified schema.
- This is to ensure a consistent schema being used for the streaming query.

File Source

File Source Implementation:

```
# Import Spark Session
# Import split from the pyspark.sql.functions
# Import StructType, StringType, and StructField from
   # pyspark.sql.types
# Create Spark Session
<your schema> = StructType([StructField(\
          “<column name>”, StringType(), True), \
          .....................
          ]) # Create schema
```

File Source

```
lines = <spark session>.readStream.format(“csv”).\
       option(“maxFilesPerTrigger”, 1).\
       option(“header”, True).\
       option(“path”, “<path to your folder>”).\
       schema(<your schema>).load()
```

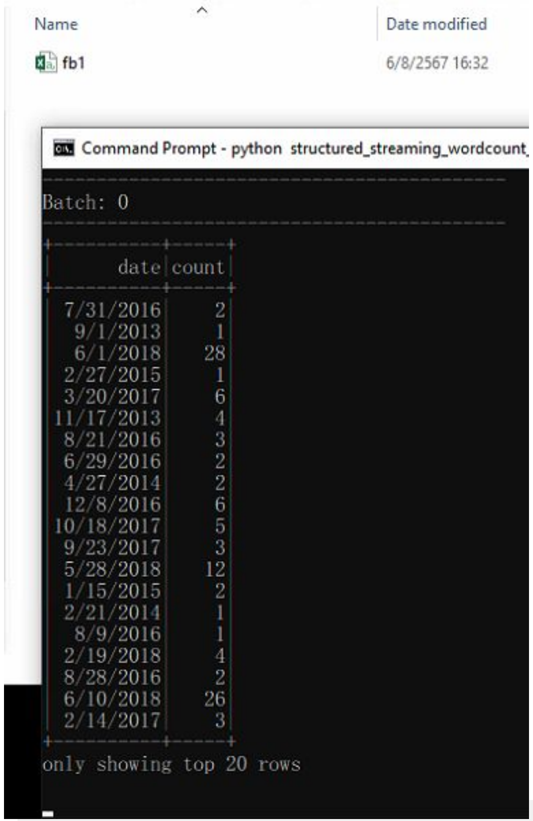
- maxFilesPerTrigger is the maximum number of new files to be considered in every trigger.
- The option path is used to identify the folder containing data.

File Source

```
# Print schema
words = lines.withColumn(“date”, \
          split(lines[“status_published”], “ ”).\
          getItem(1)) # Get the date of status published
wordCounts = words.groupBy(“date”, \
             “status_type”).count() # Count based on date
                                   # and status type
# Write Stream
# Wait for termination
```

File Source

- Create a folder and add fb1.csv into the created folder
- Run the Stream File Source Python code.



Name | Date modified

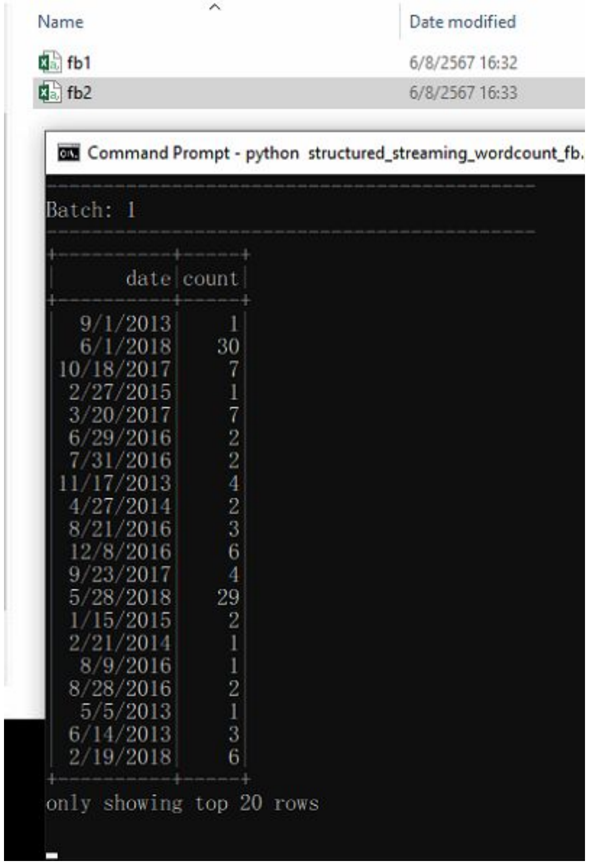
fb1 | 6/8/2567 16:32

Command Prompt - python structured_streaming_wordcount_

```
-------------------------------------------
Batch: 0
-------------------------------------------
+----------+-----+
|      date|count|
+----------+-----+
| 7/31/2016|    2|
|  9/1/2013|    1|
|  6/1/2018|   28|
| 2/27/2015|    1|
| 3/20/2017|    6|
|11/17/2013|    4|
| 8/21/2016|    3|
| 6/29/2016|    2|
| 4/27/2014|    2|
| 12/8/2016|    6|
|10/18/2017|    5|
| 9/23/2017|    3|
| 5/28/2018|   12|
| 1/15/2015|    2|
| 2/21/2014|    1|
|  8/9/2016|    1|
| 2/19/2018|    4|
| 8/28/2016|    2|
| 6/10/2018|   26|
| 2/14/2017|    3|
+----------+-----+
only showing top 20 rows
```

File Source

- Add the file fb2.csv into the created folder
- You will see the second Dataframe, where the numbers of counts are updated.



Name	Date modified
fb1	6/8/2567 16:32
fb2	6/8/2567 16:33

Command Prompt - python structured_streaming_wordcount_fb.

```
-------------------------------------------
Batch: 1
-------------------------------------------
+----------+-----+
|      date|count|
+----------+-----+
|  9/1/2013|    1|
|  6/1/2018|   30|
|10/18/2017|    7|
| 2/27/2015|    1|
| 3/20/2017|    7|
| 6/29/2016|    2|
| 7/31/2016|    2|
|11/17/2013|    4|
| 4/27/2014|    2|
| 8/21/2016|    3|
| 12/8/2016|    6|
| 9/23/2017|    4|
| 5/28/2018|   29|
| 1/15/2015|    2|
| 2/21/2014|    1|
|  8/9/2016|    1|
| 8/28/2016|    2|
|  5/5/2013|    1|
| 6/14/2013|    3|
| 2/19/2018|    6|
+----------+-----+
only showing top 20 rows
```

Assignment (2 points)

- Please implement the socket and file codes.
- Please execute your code and show the results to get 2 points.

References

- Chambers, B., & Zaharia, M. (2018). *Spark: The definitive guide: Big data processing made simple.* " O'Reilly Media, Inc.".
- Antolínez García, A. (2023). *Hands-on Guide to Apache Spark 3: Build Scalable Computing Engines for Batch and Stream Data Processing*. Berkeley, CA: Apress.
- Spark. https://spark.apache.org. Accessed: 2023-10-16.