```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "# Lab 4 Spark Streaming \n",
    "\n",
    "This lab guides you through the process of setting up Java and Spark on your local machine, and running a simple Spark Streaming application in Python. The application will count the frequency of words received on a TCP port over a one-minute window, with batch intervals of 1 second.\n"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "## Step 1: Install Java\n",
    "\n",
    "Apache Spark requires Java. Follow these steps to install Java:\n",
    "### Windows\n",
    "\n",
    "1. **Download Java**: Visit the [Oracle website](https://www.oracle.com/java/technologies/javase-jdk11-downloads.html) and download the Java JDK.\n",
    "2. **Install Java**: Follow the installation instructions for your operating system.\n",
    "3. **Verify Installation**: Open a terminal and run `java -version`. You should see the version of Java you installed. If you see an error, you may need to add Java to your PATH.\n",
    "4. **Add Java to PATH**: If you see an error when running `java -version`, you may need to add Java to your PATH. Follow these instructions to add Java to your PATH:\n",
    "    - Right click on \"My Computer\" or \"This PC\" and select \"Properties\".\n",
    "    - Click \"Advanced system settings\".\n",
    "    - Click \"Environment Variables\".\n",
    "    - Under \"System variables\", click \"New\" to create a new variable.\n",
    "    - For the variable name, enter `JAVA_HOME`.\n",
    "    - For the variable value, enter the path to your Java installation. For example, `C:\\Program Files\\Java\\jdk-11.0.7`.\n",
    "    - Click \"OK\" to apply the changes.\n",
    "\n",
    "### Linux\n",
```

```
    "1. **Open a terminal**: and update the package repository to ensure
you have the latest version of the software:\n",
    "``` bash\n",
    "sudo apt-get update\n",
    "```\n",
    "\n",
    "2. **Install Java**: Run the following command in your terminal to
install Java:\n",
    "``` bash\n",
    "sudo apt-get install default-jdk\n",
    "```\n",
    "3. **Verify Installation**: Run the following command in your terminal
to verify the installation:\n",
    "``` bash\n",
    "java -version\n",
    "```\n",
    "You should see the version of Java you installed.\n",
    "4. **Set JAVA_HOME Environment Variable**: Find out where Java is
installed on your system by running the following command in your
terminal:\n",
    "``` bash\n",
    "update-alternatives --config java\n",
    "```\n",
    "- Copy the installation path of Java.\n",
    "- Open your `.bashrc` file in a text editor. You can use the following
command to open the file in the nano text editor:\n",
    "``` bash\n",
    "nano ~/.bashrc\n",
    "```\n",
    "- Add the following line to the end of the file, replacing
`/path/to/java` with the installation path of Java:\n",
    "``` bash\n",
    "export JAVA_HOME=/path/to/java\n",
    "```\n",
    "- Save the file and exit the text editor.\n",
    "- Run the following command in your terminal to apply the changes:\n",
    "``` bash\n",
    "source ~/.bashrc\n",
    "```\n",
    "\n",
    "  \n",
    "## Mac\n",
    "``` bash\n",
    "brew install openjdk\n",
    "```\n",
    "To run this command, you must have homebrew installed on your Mac. If
```

```
you do not have it installed, you can go to the [homebrew
website](https://brew.sh/)."
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "## Step 2: Install Spark\n",
    "\n",
    "Now, install Apache Spark:\n",
    "### Windows\n",
    "\n",
    "1. **Download Spark**: Go to the [Apache Spark
website](https://spark.apache.org/downloads.html) and download Spark. You
can choose the latest version of Spark pre-built for Apache Hadoop. \n",
    "2. **Unzip the Spark Package**: Extract the downloaded file to a
desired location. (i.e. c:\\spark)\n",
    "3. **Set Environment Variables**: We need to set `SPARK_HOME`
environment variable to the location where Spark is installed. Similar to
setting Java to PATH, follow these steps:\n",
    "For Windows:\n",
    "    - Right click on \"My Computer\" or \"This PC\" and select
\"Properties\".\n",
    "    - Click on \"Advanced system settings\".\n",
    "    - Click on \"Environment Variables\".\n",
    "    - Under \"System variables\", click \"New\" to create a new
variable.\n",
    "    - For the variable name, enter `SPARK_HOME`.\n",
    "    - For the variable value, enter the location where Spark is
installed (i.e. `c:\\spark`).\n",
    "    - Click \"OK\" to apply the changes.\n",
    "    - Now to add Spark to the PATH, in the \"System variables\"
section, find the \"Path\" variable and click \"Edit\".\n",
    "    - Click \"New\" and enter `%SPARK_HOME%\\bin`.\n",
    "    - Click \"OK\" to apply the changes.\n",
    "\n",
    "4. **Install winutils.exe and hadoop.dll**. Spark on Windows requires
Hadoop winutils to run. \n",
    "   - You can download it from
[here](https://github.com/steveloughran/winutils/tree/master), under the
`hadoop-3.0.0` folder.\n",
    "   - Create a hadoop directory on your system, for example
`C:\\hadoop\\bin` and copy the `winutils.exe` file into
`C:\\hadoop\\bin`.\n",
    "   - Set the `HADOOP_HOME` environment variable to the location of the
```

```
`hadoop` directory. (Follow the same steps as setting `SPARK_HOME`).\n",
    "      - Variable name: `HADOOP_HOME`\n",
    "      - Variable value: `C:\\hadoop`\n",
    "      - Click \"OK\" to apply the changes.\n",
    "    - Now to add Hadoop to the PATH, in the \"System variables\"
section, find the \"Path\" variable and click \"Edit\".\n",
    "     - Click \"New\" and enter `%HADOOP_HOME%\\bin`.\n",
    "     - Click \"OK\" to apply the changes.\n",
    "   - Test the installation by opening command prompt and running
`winutils.exe ls /` and you should see a list of files and folders.\n",
    "\n",
    "5. **Verify Installation**: Run `spark-shell` in the terminal. If you
see the message `Welcome to Spark`, then Spark is installed correctly.\n",
    "\n",
    "\n",
    "### Linux\n",
    "1. **Download Spark**: Go to the [Apache Spark
website](https://spark.apache.org/downloads.html) and download Spark. You
can choose the latest version of Spark pre-built for Apache Hadoop.\n",
    "2. **Unzip the Spark Package**: Extract the downloaded file using the
following command in your terminal:\n",
    "``` bash\n",
    "tar -xvf spark-file-name.tgz \n",
    "```\n",
    "- Replace `spark-file-name.tgz` with the name of the file you
downloaded.\n",
    "3. **Set Environment Variables**: We need to set `SPARK_HOME`
environment variable to the location where Spark is installed. Follow these
steps to set the environment variable:\n",
    "    - Open your `.bashrc` file in a text editor. You can use the
following command to open the file in the nano text editor:\n",
    "    ``` bash\n",
    "    nano ~/.bashrc\n",
    "    ```\n",
    "    - Add the following line to the end of the file, replacing
`/path/to/spark` with the location where Spark is installed:\n",
    "    ``` bash\n",
    "    export SPARK_HOME=/path/to/spark\n",
    "    ```\n",
    "    - Still in the profile settings file, add the following line:
\n",
    "    ``` bash\n",
    "    export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin\n",
    "    ```\n",
    "    - Save the file and exit the text editor.\n",
    "    - Run the following command in your terminal to apply the
```

```
changes:\n",
    "    ``` bash\n",
    "    source ~/.bashrc\n",
    "    ```\n",
    "\n",
    "4. **Verify Installation**: Run `spark-shell` in the terminal. If you see the message `Welcome to Spark`, then Spark is installed correctly.\n",
    "\n",
    "\n",
    "### Mac\n",
    "``` bash\n",
    "brew install apache-spark\n",
    "```\n",
    "Once installed, run `spark-shell` in the terminal. If you see the message `Welcome to Spark`, then Spark is installed correctly.\n"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "## Step 3: Python Dependencies\n",
    "\n",
    "Install PySpark and findspark, which will allow us to use Spark in Python:\n",
    "\n",
    "```bash\n",
    "pip install pyspark findspark\n"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "## Step 4: Spark Streaming Application\n",
    "\n",
    "We will create a Spark Streaming application in Python to count words received on a TCP port.\n",
    "Paste the following code in a separate Python file, then run the application.\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {},
```

```
   "outputs": [],
   "source": [
    "from pyspark import SparkConf, SparkContext\n",
    "from pyspark.streaming import StreamingContext\n",
    "\n",
    "# Create a SparkConf and SparkContext\n",
    "conf =
SparkConf().setMaster(\"local[2]\").setAppName(\"NetworkWordCount\")\n",
    "\n",
    "# Create a local StreamingContext with batch interval of 1 second
using the SparkContext\n",
    "ssc = StreamingContext(SparkContext(conf=conf), 1)\n",
    "\n",
    "# Create a DStream that connects to localhost:9999\n",
    "lines = ssc.socketTextStream(\"localhost\", 9999)\n",
    "\n",
    "# Split each line into words\n",
    "words = lines.flatMap(lambda line: line.split(\" \"))\n",
    "\n",
    "# Count each word in each batch\n",
    "pairs = words.map(lambda word: (word, 1))\n",
    "wordCounts = pairs.reduceByKey(lambda x, y: x + y)\n",
    "\n",
    "# Print the first ten elements of each RDD generated in this DStream
to the console\n",
    "wordCounts.pprint()\n",
    "\n",
    "ssc.start()             # Start the computation\n",
    "ssc.awaitTermination()  # Wait for the computation to terminate\n"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "## Step 5: Running the Application\n",
    "\n",
    "After running the above Python code in your Jupyter Notebook, start a
TCP server on port 9999 and send some text data. You should see the word
counts being updated every second for a window of one minute.\n"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
```

```
    "1. Start a TCP server on port 9999. Anything you type in this terminal will be sent to this port:\n",
    "``` bash\n",
    "nc -lk 9999\n",
    "```\n",
    "*For Mac users: please first install nmap and use ncat instead...*\n",
    "``` bash\n",
    "brew install nmap\n",
    "ncat -l -k 9999\n",
    "```\n",
    "2. This starts the python program \n",
    "3. Type some text in the terminal where you started the TCP server. You should see the word counts being updated every second for a window of one minute in the python program. As a example, try typing \"hello world\" and see the word counts being updated in the python program. Now try typing \"hello world hello\" and see what the output gives you."
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": []
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.10.11"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 2
}
```