INP7079233 - BIG DATA COMPUTING (Proff. A: Pietracaprina and F. Silvestri) 2022-2023

Home / My courses / 2022-IN2547-003PD-2022-INP7079233-G2GR1 / Homework 1 / Machine Setup

Machine Setup

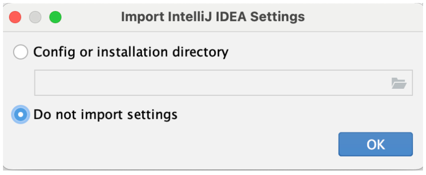
Before doing any work, you should setup your machine. First of all, you need to have the Java Development Kit (JDK) version 8 installed on your machine. If you do not, head to Oracle download page and download the Java Development Kit version 8. Version 9 (or later versions) might give problems, so we should avoid them for the time being.

Quick links:

- Instructions for Java users
- Instructions for Python users

Instructions for Java users

1. Create a directory **BDC** on your computer which will be the main directory for your homeworks. In this directory, put the file build.gradle which you must download here.
2. Install Intellij Idea (Community edition), version 2022.3 on your system from this download page. (For an installation guide you can look at the official install and set-up page.)
3. After installation is completed, you must configure Intellij for a first run. Launch Intellij. After the initial steps for accepting their User Agreement and Data Sharing options, follow these instructions:
    1. Choose "Do not import settings"

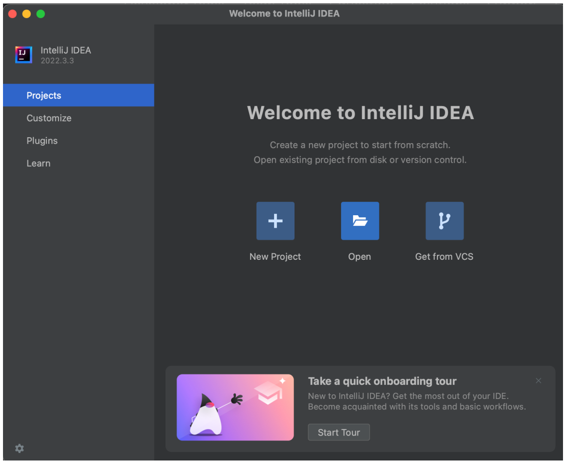
       

       Import IntelliJ IDEA Settings

       ○ Config or installation directory

       ● Do not import settings

    2. Select Open and use the file selection dialog that pops up to select the build.gradle file contained in the directory you created in Step 1

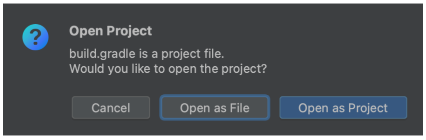
       Welcome to IntelliJ IDEA

       Create a new project to start from scratch.
       Open existing project from disk or version control.

       Take a quick onboarding tour

       New to IntelliJ IDEA? Get the most out of your IDE.
       Become acquainted with its tools and basic workflows.

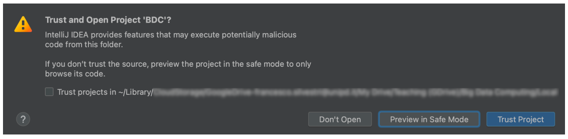
    3. Click "Open as Project"

       Open Project

       build.gradle is a project file.
       Would you like to open the project?

    4. Flag the checkbox "Trust project in ..." (you will see the gradle URL) and then the "Trust Project" button.

       Trust and Open Project 'BDC'?

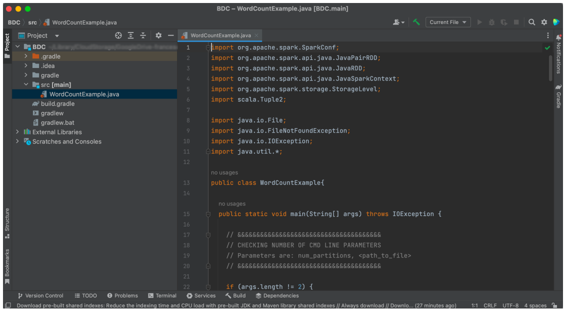
       IntelliJ IDEA provides features that may execute potentially malicious code from this folder.

       If you don't trust the source, preview the project in the safe mode to only browse its code.

       □ Trust projects in ~/Library/[redacted]

    5. You will have to wait a couple of minutes until Intellij configures itself.

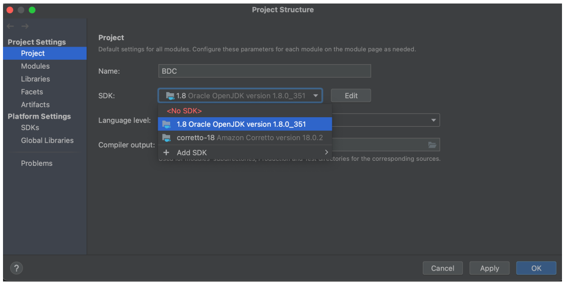
4. Create a directory **BDC/src/**. As a default, put all of your programs in **BDC/src/**, and all datasets that you want to provide as inputs to your programs, in the root directory **BDC/**.
5. Use the project navigation panel on the left to open the files (e.g., programs, datasets, etc.). To test, open a java program (for instance copy in BDC/src the file WordCountExample.java and in BDC/ the input file sentence_small.txt) as in the following screenshot:



```
import org.apache.spark.SparkConf;
import org.apache.spark.api.java.JavaPairRDD;
import org.apache.spark.api.java.JavaRDD;
import org.apache.spark.api.java.JavaSparkContext;
import org.apache.spark.storage.StorageLevel;
import scala.Tuple2;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.*;

no usages
public class WordCountExample{

  no usages
  public static void main(String[] args) throws IOException {

    // &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
    // CHECKING NUMBER OF CMD LINE PARAMETERS
    // Parameters are: num_partitions, <path_to_file>
    // &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

    if (args.length != 2) {
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK and Maven library shared indexes // Always download // Downlo... (27 minutes ago) 1:1 CRLF UTF-8 4 spaces

6. Check if you are using the correct JDK. Open the menu File/Project Structure and select Project from the left panel. Then, open the drop-down menu and select JDK version 8, as in the figure above (your list might differ).



Project

Default settings for all modules. Configure these parameters for each module on the module page as needed.

Name: BDC

SDK: 1.8 Oracle OpenJDK version 1.8.0_351

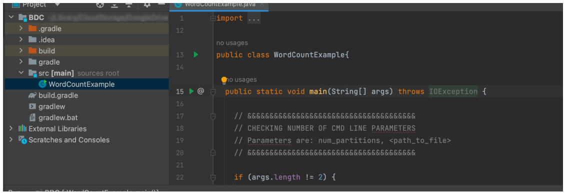
- 1.8 Oracle OpenJDK version 1.8.0_351
- corretto-18 Amazon Corretto version 18.0.2

Language level:

Compiler output:

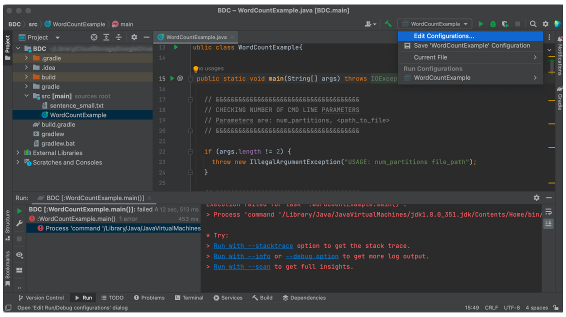
...ctories for the corresponding sources.

7. On the line of the **main** method there is a green arrow, which allows you to run your code.



```
import ...

no usages
public class WordCountExample{

  no usages
  public static void main(String[] args) throws IOException {

    // &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
    // CHECKING NUMBER OF CMD LINE PARAMETERS
    // Parameters are: num_partitions, <path_to_file>
    // &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

    if (args.length != 2) {
```

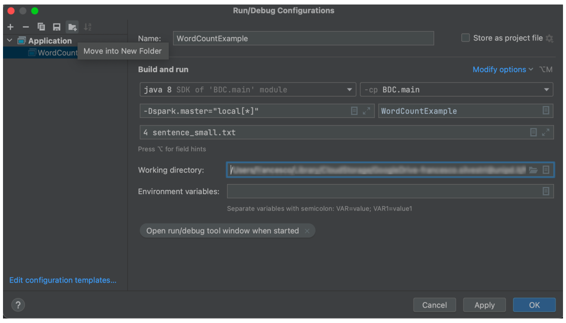
8. Clicking the green arrow will compile the code and run it. The run will not succeed (exit code 1 at the bottom of the screen) since you must configure a set of execution parameters. To do so, use the drop-down menu on the top-right of the Intellij window, where the name of the program (WordCountExample, in the example) appears, and select **Edit Configurations** to get a dialog window which you must fill as indicated in the following images.



```
ublic class WordCountExample{

  no usages
  public static void main(String[] args) throws IOExcep

    // &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
    // CHECKING NUMBER OF CMD LINE PARAMETERS
    // Parameters are: num_partitions, <path_to_file>
    // &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

    if (args.length != 2) {
      throw new IllegalArgumentException("USAGE: num_partitions file_path");
    }
```

```
BDC [:WordCountExample.main()]: failed  A 12 sec, 513 ms
  :WordCountExample.main()  1 error  453 ms
    Process 'command '/Library/Java/JavaVirtualMachines

> Process 'command '/Library/Java/JavaVirtualMachines/jdk1.8.0_351.jdk/Contents/Home/bin/

* Try:
> Run with --stacktrace option to get the stack trace.
> Run with --info or --debug option to get more log output.
> Run with --scan to get full insights.
```

Open 'Edit Run/Debug configurations' dialog 15:49 CRLF UTF-8 4 spaces



Name: WordCountExample

java 8 SDK of 'BDC.main' module

-cp BDC.main

-Dspark.master="local[*]"

WordCountExample

4 sentence_small.txt

Working directory: [redacted]

Environment variables:

9. Note that the VM options field (where the spark.master property is set) may be hidden, and must be retrieved by clicking on the Modify options blue text. Specifically, VM options specify that you want to run Spark in local mode (spark.master is a java system property), while CLI arguments to your application are the arguments passed to the main method (in this example, the line with the string "4 sentence_small.txt" encodes the arguments for the main in WordCountExample.java).

Instructions for Python users

These instructions assume that Python is installed in your machine. First of all, one has to download the Spark APIs for Python to be able to import its modules and define a Spark session. A possible way to do it will be presented in the following.

Step 1: Download and install a JDK

Since Pyspark will translate the Python instructions into Java bytecodes that will be executed in the JVM of each node of the cluster, a Java Runtime Environment must be available throughout the system with its compiler and its APIs. Head to Oracle download page and download the Java Development Kit version 8. Version 9 (or later versions) might give problems, so we should avoid them for the time being. **Please take notice of the path to the directory where the JDK is installed** (usually something like "/usr/lib/jvm/*"). It will be needed in Step 3 to set the JAVA_HOME environmental variable.

Step 2: Use pip

In order to download the Spark libraries, it is sufficient to open a terminal and to type

```
$ pip install pyspark
```

This will also take care of installing the dependencies (e.g. py4j), and will tell you the directory where Pyspark is installed. Remark: if conda is installed, one can equivalently use its package manager, writing the command

```
$ conda install pyspark
```

Please take notice of the path to the directory where Pyspark is installed. You will need it in Step 3 to set the SPARK_HOME environmental variable. If Anaconda is present on the computer, then both its package manager and pip will install pyspark in one of its subdirectories.

Step 3: Configure the environment variables

As an intermediate passage it is required to find the installation directory of Spark and Java.

By default, under Unix-based systems, Java and Pyspark should be respectively found under "/usr/lib/jvm/*" and "/usr/lib/python*/dist-packages/pip". If Anaconda is present on the computer, then both its package manager and pip will install pyspark in one of its subdirectories.

Linux and OS x instructions

Run in a terminal

```
$ nano ~/.bashrc
```

and add the following lines to the file

```
export JAVA_HOME="/path/to/JDK/"
export SPARK_HOME="/path/to/Spark/"
export PATH=$JAVA_HOME/bin:$SPARK_HOME:$SPARK_HOME/bin:$PATH
export PYTHONPATH=$SPARK_HOME/python:$SPARK_HOME/python/build:$PYTHONPATH
```

In order to make the changes have effect, one can just close and reopen the terminal or give the command

```
$ source ~/.bashrc
```

Windows instructions

Under Windows the procedure is slightly different:

- Download the Windows binaries for Hadoop from this link.
- Open the Windows tool to modify the environment variables (it can be found searching for such variables in the start panel)
- Add "path_to_Hadoop\bin" to the environment variable PATH
- Create a new variable HADOOP_HOME with value "path_to_Hadoop".
- Create a new variable SPARK_HOME with value "path_to_Pyspark". You can find this path looking for the folder where the library is installed. If you installed pyspark with conda it should be in "C:\Users\<User>\anaconda3\Lib\site-packages\pyspark", if you installed it without conda is in "<folder where python is installed>\Lib\site-packages\pyspark", by default python is installed in "C:\Users\<User>\AppData\Python".
- You may need to restart the machine to make sure that the changes are applied.

Step 4: Verify the installation

If the installation ended properly, the call of Pyspark in the terminal

```
$ pyspark
```

will show something like

```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.3.0
      /_/
Using Python version 3.6.4 (default, Jan 16 2018 18:10:19)
SparkSession available as 'spark'.
>>>
```

Usage example

The actual execution of a program MyProgram.py on some program-parameters will be made using the command

```
$ python path-to-program/MyProgram.py program-parameters
```

As a test, you can run the program WordCountExample.py on the input file sentence_small.txt

Appendix: configuration of Jupyter

It is possible to automatically create the Spark context and to use a Jupyter notebook, if already installed, with the command

```
$ PYSPARK_DRIVER_PYTHON=jupyter PYSPARK_DRIVER_PYTHON_OPTS=notebook pyspark
```

The linking can also be made permanent adding to the ~/.bashrc the lines

```
export PYSPARK_DRIVER_PYTHON="jupyter"
export PYSPARK_DRIVER_PYTHON_OPTS="notebook"
```

Last modified: Friday, 24 March 2023, 5:11 PM

◄ Instructions to use the VMs

Jump to...

Assignment of Homework 1 (DEADLINE April 12, 23:59) ►