**Number of Applications on Multiple Computers**

-Ravi Kumar Enukonda

1. **Overview and design:**

**Problem statement:**

The Java implementation calculates the ***minimum number of copies of applications*** required to install on multiple computers based on below scenario.

“each copy of the application (ID XXX) allows the user to install the application on to two computers if at least one of them is a laptop” where the input records are provided from a CSV file in the format below.

*ComputerID,UserID,ApplicationID,ComputerType,Comment*

*1,1,374,LAPTOP,Exported from System A*

*2,2,374,DESKTOP,Exported from System A*

**Design and implementation:**

I have created a Maven project and developed the solution in IntelliJ IDE. Predominantly I have used **JAVA 8 concepts like streams and lambda expressions**. By using java streams I have improved the performance very significantly in processing large CSV file (1GB). Written number of unit test cases to check the functionality of the application and various methods.

I have used Maven *Lombok* to reduce development time, keep the code clean to avoid writing/generating the setters and getter manually. I have created a *enum* ComputerType*,* to differentiate LAPTOP and DESKTOP types.

As part of the functionality implementation, business logic is implemented in three classes, *ProcessInputData.java, Compute.java and TotalApps.java*. Created model classes as per the need. Created different test set CSV files under resources directory for unit testing.

**Performance at a glance:**

Time taken to process ***sample-small.csv*** is **~0.5 seconds** ( 10 MB )

Time taken to process ***sample-large.csv*** is **~53 seconds** ( 1.03GB)

***Note:*** I have used 2019 MacBook pro to develop, build and run the project with 2.7 GHz Inter Core i7 and 16GB RAM.



1. **Build the code, generate executable Jar file and run it:**
2. Clone the Git repository “<https://github.com/ravienu/AppsOnComputers.git>”

git clone https://github.com/ravienu/AppsOnComputers.git

1. Navigate to the directory “…/AppsOnComputers/AppsToInstall”
2. Open the command prompt and execute the commands below

**mvn install**

**mvn clean compile assembly:single**

The above command will generate the executable jar file “*AppsToInstall-1.0.jar*” in the directory “…\AppsToInstall\target”

**Note:** I have used *maven-assembly-plugin* togenerate executable JAR file with all the dependencies.

1. Run the Jar file.

Navigate to the directory “…\AppsToInstall\target”

**java -jar -Xmx512m AppsToInstall-1.0.jar <Path\_To\_CSV\_File>**

**Usage:**

**java -jar -Xmx512m AppsToInstall-1.0.jar “…/sample-small.csv”**

**java -jar -Xmx5120m AppsToInstall-1.0.jar “…/sample-large.csv”**

The highlighted option (-Xmx512m / -Xmx5120m) in blue colour in above commands is, maximum heap memory for a java process. (It can be increased or omitted based on need)

For the sample-large.csv, allocation of 5120MB worked well. Make sure the system you are running in has more than 5120MB of free space for sample-large.csv. In case If you get “OutOfMemoryError”, increase the heap size from ~500MB to 1000MB (**-Xmx6000m**).

**Note:** sample-large.csv file provided is not included as part of the jar file resources due to size constraint.

Example Run:

Ravis-MacBook-Pro:target ravikumarenukonda$ java -jar -Xmx512m AppsToInstall-1.0.jar "/Volumes/RaviData/Ravi\_Mac/multiuserapp/src/main/resources/sample-small.csv

**Output:**

Number of copies of applications required : 110069

Ravis-MacBook-Pro:target ravikumarenukonda$ java -jar -Xmx5120m AppsToInstall-1.0.jar "/Volumes/RaviData/Ravi\_Mac/multiuserapp/src/main/resources/sample-large.csv"

**Output:**

Number of copies of applications required : 10091137