Map Reduce implementation

Vũ Đinh Anh - m21.ict.001

Nguyễn Lê Tuấn Duy - m21.ict.004

Instructions to install and run

Java 11 was used to complete this project. Therefore, to have better performance and experience, Java 11 should be used.

```
openjdk 11.0.14 2022-01-18
OpenJDK Runtime Environment (build 11.0.14+9-Ubuntu-0ubuntu2.20.04)
OpenJDK 64-Bit Server VM (build 11.0.14+9-Ubuntu-0ubuntu2.20.04, mixed mode, sharing)
```

To compile, it's quite straightforward. One can open the folder where contains all the source code, then just run `javac *.java`. Then 3 terminal should be opened in same directory. Let name them `main`, `daemon 0`, `daemon 1`.

On `daemon 0`, run `java Daemon 0`. Do same thing with `daemon 1`.

On `main`, run `java Split data.txt 2`. Wait it complete, then now run `java Launch 2`. Finalresult.txt should be seen in the directory.

Description

- Split.java
  - Open socket to other nodes
  - Then send each chunk of a file to each node
- Launch.java
  - Invoke Daemon.call()
  - After **all** DoneCallBack.completed() receive all files, reduceMap() is called.
- Daemon.java
  - Act as ServerSocket to receive file from Split
  - Then act as RMI server
  - When Daemon.call() is invoked, while reducing file, Daemon act as ServerSocket
- DoneCallback.java
  - When executeMap() is done, DoneCallBack.completed() open socket to receive file from ServerSocket
- WordCount.java
- DaemonInterface.java
- MapReduceInterface.java
- CallBackInterface.java

Unfortunately, `notify()`, `wait()`, `synchronized` methods were NOT used to implement CallBack because we forget them exist.