File I/O Performance Measurement

This Java application measures and compares the performance of reading and writing operations for binary and text files, with and without buffering. It generates a CSV (Comma-Separated Values) file with the results.

Usage

To run the application, follow these steps:

1. Compile and pack the application.

```
./mvnw package
```

2. Execute the program. You can specify two optional command-line arguments: `max power of 2` and `interval`. `max power of 2` determines the maximum file size in bytes (2^maxPower), and `interval` sets the step size between file sizes. If no arguments are provided, default 2^{20} and 2 values will be used.

```
java -jar target/java-ios-practical-content-1.0-SNAPSHOT.jar <max
power of 2> <interval>
```

3. The application will perform the following operations for binary and text files, with and without buffering:
    - Write files of various sizes.
    - Read the previously written files.
    - Record the time taken for each operation.
4. A CSV file named `data.csv` will be generated in the current directory. This file contains the following columns:
    - `Name`: File name.
    - `Type`: File type (BINARY or TEXT).
    - `Buffered`: Whether buffering is enabled (true or false).
    - `Size`: File size in bytes.
    - `WTime`: Time taken to write the file (in milliseconds).
    - `RTime`: Time taken to read the file (in milliseconds).

Example

Here's an example of running the application with custom `maxPower` and `interval` values:

```
java -jar target/java-ios-practical-content-1.0-SNAPSHOT.jar 20 2
```

In this example, the application will measure performance for file sizes from 2^0 bytes up to 2^{20} bytes (1 megabyte), with a step increment of 2 .

Generate charts and dataTable

Prerequisites for SVG Generation: `matplotlib` and `pandas`

To generate SVG (Scalable Vector Graphics) files, you will need to install the `matplotlib` and `pandas` Python libraries. You can install them using the following `pip` commands:

```
pip install matplotlib
pip install pandas
```

After creating the `data.csv` file, you can use the following commands to create graphs and a data table:

```
python chart.py
python dataTable.py
```

These commands will help you visualize and analyze the data generated by the application.

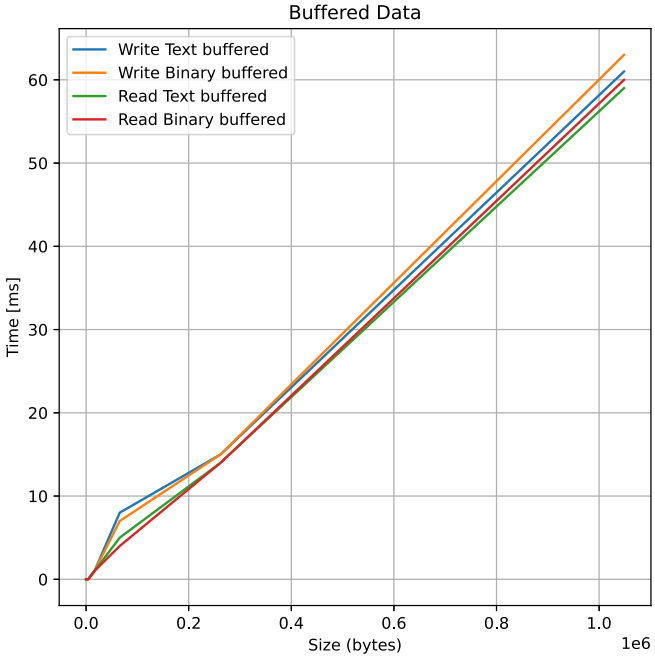
Note

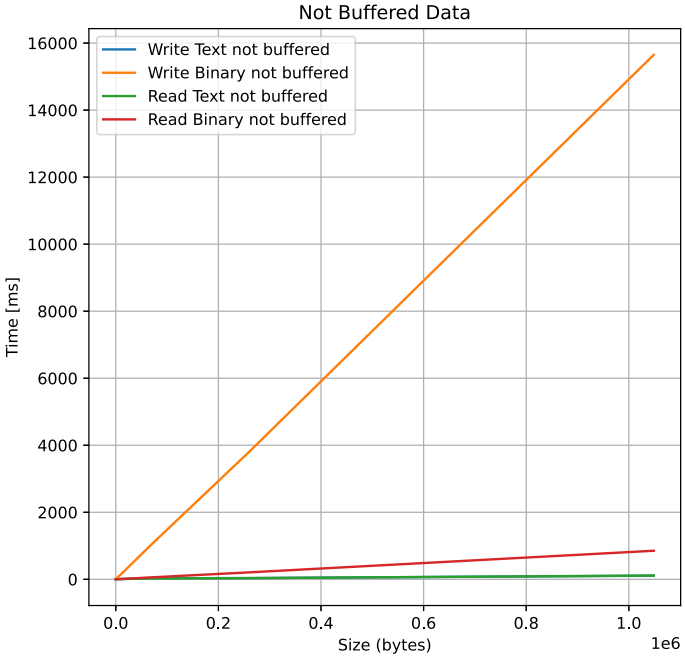
- The application uses classes and methods from the `ch.heigvd.dai` package for measuring file I/O performance.
- The `isNumber` function checks if a string can be converted to an integer.

Please make sure you have the required dependencies and permissions to run this application.

Exemple of generated datas and charts

With this project, we have generated the following data, ranging from 1 byte to 2^{20} bytes (1 megabyte) with an interval of 2. Here are the results:





Type	Buffered	Size	WTime	RTime
BINARY	False	1	0	0
BINARY	False	4	0	0
BINARY	False	16	0	0
BINARY	False	64	1	0
BINARY	False	256	4	0
BINARY	False	1024	15	1
BINARY	False	4096	63	3
BINARY	False	16384	242	13
BINARY	False	65536	981	55
BINARY	False	262144	3836	209
BINARY	False	1048576	15649	852
BINARY	True	1	0	0
BINARY	True	4	0	0
BINARY	True	16	0	0
BINARY	True	64	0	0
BINARY	True	256	0	0
BINARY	True	1024	0	0
BINARY	True	4096	0	0
BINARY	True	16384	1	1
BINARY	True	65536	7	4
BINARY	True	262144	15	14
BINARY	True	1048576	63	60
TEXT	False	1	0	1
TEXT	False	4	0	0
TEXT	False	16	0	0
TEXT	False	64	0	0
TEXT	False	256	1	1

TEXT	False	1024	4	3
TEXT	False	4096	8	3
TEXT	False	16384	12	8
TEXT	False	65536	32	24
TEXT	False	262144	37	35
TEXT	False	1048576	108	112
TEXT	True	1	0	0
TEXT	True	4	0	0
TEXT	True	16	0	0
TEXT	True	64	0	0
TEXT	True	256	0	0
TEXT	True	1024	0	0
TEXT	True	4096	0	0
TEXT	True	16384	1	1
TEXT	True	65536	8	5
TEXT	True	262144	15	14
TEXT	True	1048576	61	59