Matthieu Villard
https://github.com/matthieuVillard/RES_Labo01.git

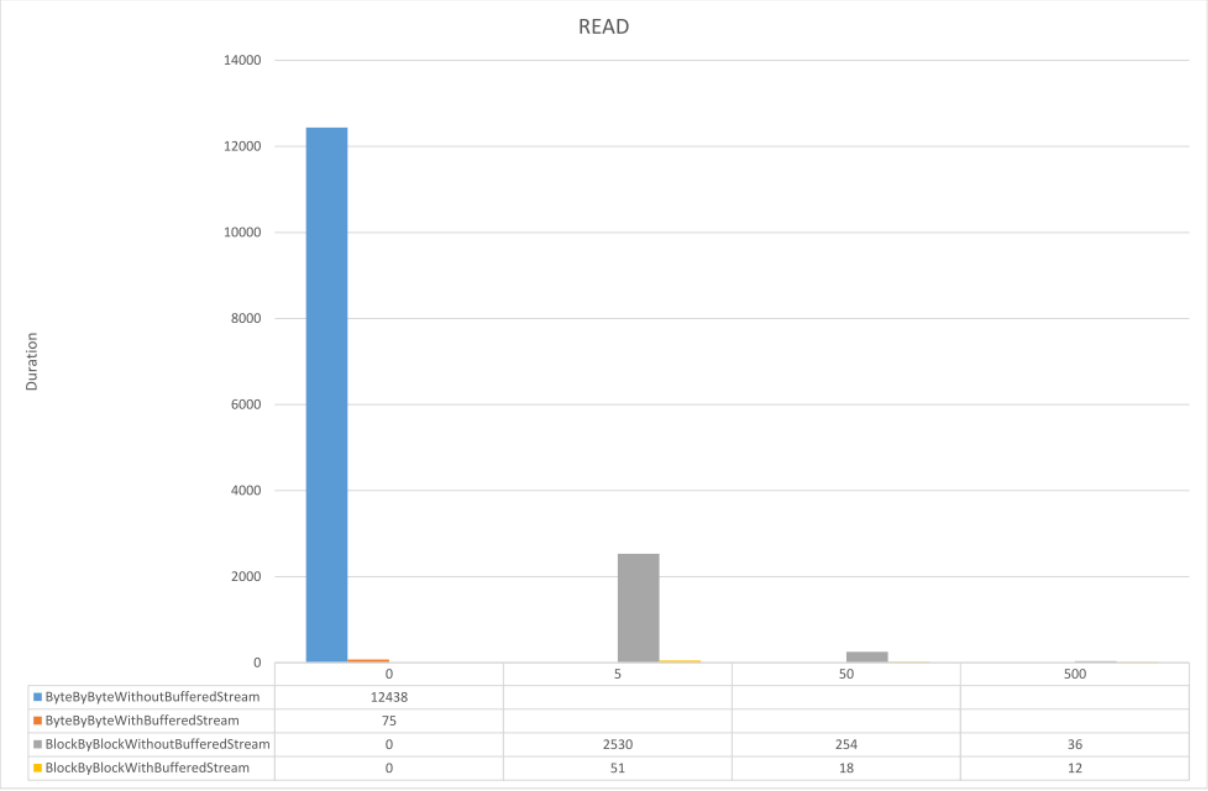
Conditions de l'expérience

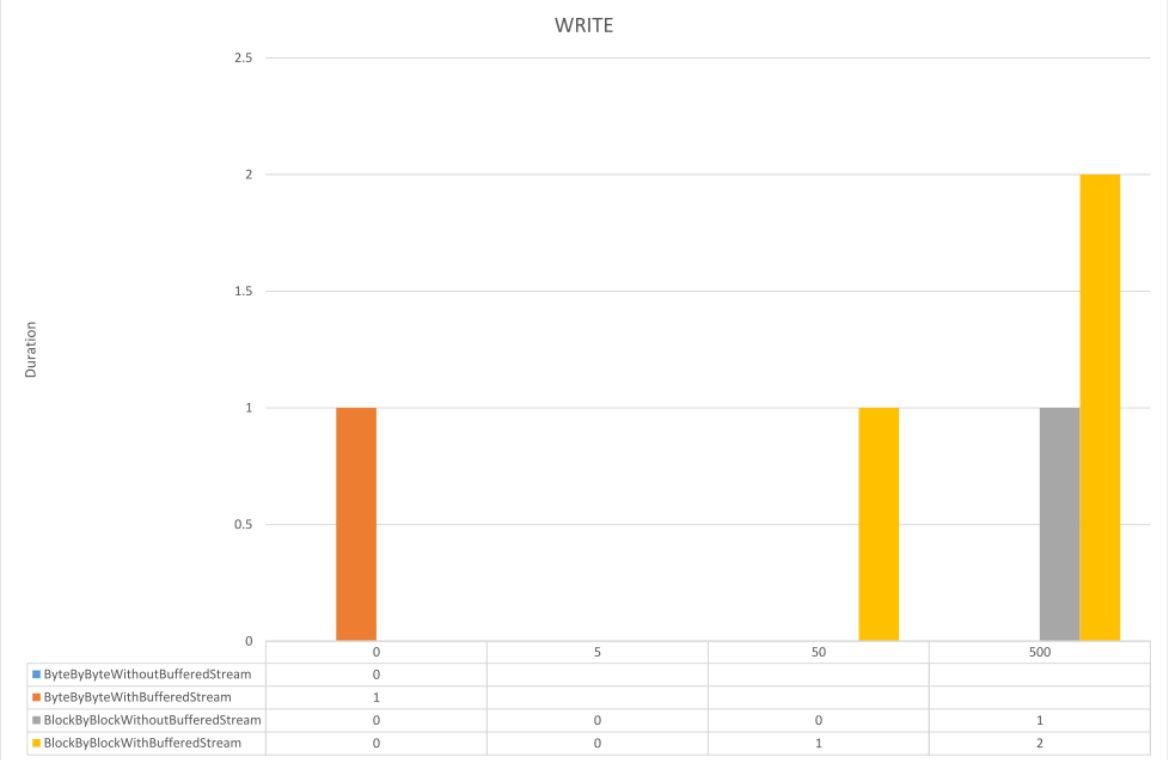
- IDE IntelliJIDEA 15.0.3
- Affichage dans la console et enregistrement dans le fichier out/result.csv
- Formatage du csv avec « ; » comme séparateur et non « , »

Mesures

operation	strategy	blockSize	fileSizeInBytes	durationInMs
WRITE	BlockByBlockWithBufferedStream	500	10485760	1
WRITE	BlockByBlockWithBufferedStream	50	10485760	0
WRITE	BlockByBlockWithBufferedStream	5	10485760	0
WRITE	ByteByByteWithBufferedStream	0	10485760	1
WRITE	BlockByBlockWithoutBufferedStream	500	10485760	0
WRITE	BlockByBlockWithoutBufferedStream	50	10485760	1
WRITE	BlockByBlockWithoutBufferedStream	5	10485760	0
WRITE	ByteByByteWithoutBufferedStream	0	10485760	0
READ	BlockByBlockWithBufferedStream	500	10485760	44
READ	BlockByBlockWithBufferedStream	50	10485760	24
READ	BlockByBlockWithBufferedStream	5	10485760	41
READ	ByteByByteWithBufferedStream	0	10485760	70
READ	BlockByBlockWithoutBufferedStream	500	10485760	35
READ	BlockByBlockWithoutBufferedStream	50	10485760	355
READ	BlockByBlockWithoutBufferedStream	5	10485760	2960
READ	ByteByByteWithoutBufferedStream	0	10485760	12910

Analyse des mesures





Il est possible de constater que de manière générale, le temps augmente lorsque la taille du bloque augmente lors de l'écriture. La lecture, au contraire de l'écriture nécessite moins de temps lorsque la taille du bloque augmente. De plus, pour la lecture, l'utilisation d'un tampon avec une lecture bloque par bloque est optimale lorsque la taille du bloque devient suffisamment grande. Pour l'écriture, lorsque la taille du bloc devient suffisamment grande, l'optimum est d'utiliser une écriture bloque par bloque sans tampon.

Modifications

Les nouvelles classes *CsvSerializer, FileRecorder et ExperimentData* ont été réalisées.

- La classe ExperimentData permet de stocker les données d'une expérience.
- La classe CsvSerializer permet, d'à partir des données d'une expérience, de les convertir en une ligne csv et de l'ajouter au flux désiré.
- La classe FileRecorder ouvre un flux sur le fichier désiré et permet ainsi d'enregistrer les données des expériences en appelant la sérialisation csv.

Dans la classe principale, il a fallu créer une instance de FileRecord pour pouvoir enregistrer les informations relatives aux expériences et appeler l'enregistrement lors de l'affichage dans la console. A noter que cela est fait différemment pour la lecture afin de pouvoir récupérer la taille totale.

Fin de la méthode `produceTestData` (écriture)

```
LOG.log(Level.INFO, "  > Done in {0} ms.", Timer.takeTime());

ExperimentData data = new ExperimentData("WRITE", ioStrategy, numberOfBytesToWrite, blockSize, Timer.takeTime());
recorder.record(data);
```

Fin de la méthode `consumeDataFromStream`(lecture)

```
LOG.log(Level.INFO, "Number of bytes read: {0}", new Object[]{totalBytes});
ExperimentData data = new ExperimentData("READ", ioStrategy, totalBytes, blockSize, Timer.takeTime());
recorder.record(data);
```