Akesson Henrik
Github: https://github.com/AkessonHenrik/LaboBenchmark

RES - Laboratoire 2

Conditions

Dans le cadre du cours sur les flux d'entrées-sorties,

Le programme a été tourné sur un ordinateur avec les spécifications suivantes :

Procésseur : Intel Core i7-6700HQ @ 2.60GHz

RAM : 16GB

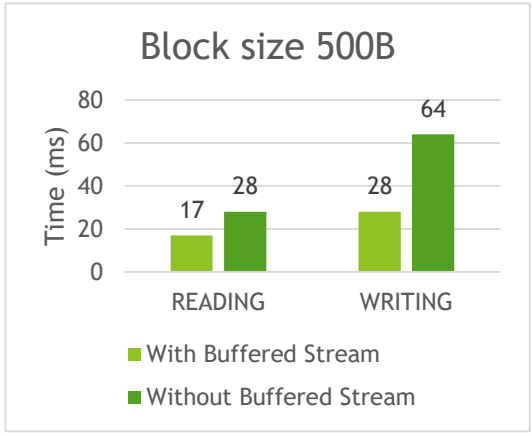
Disque dur SSD

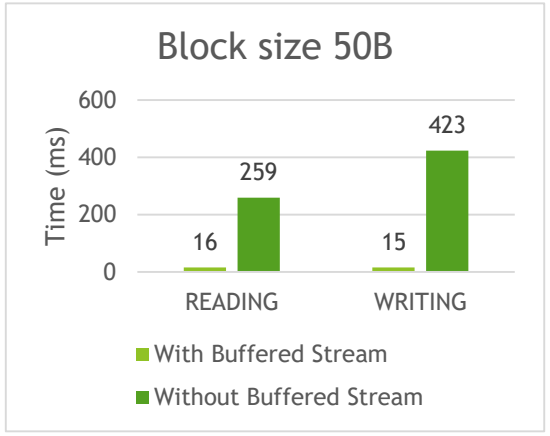
Tous les fichiers générés et lus font 10MB

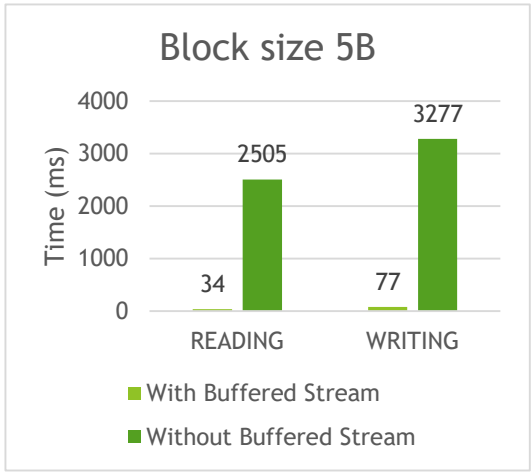
Modification du code

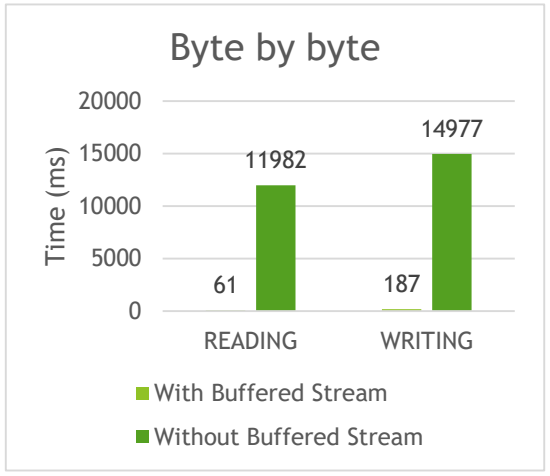
- Utilisation de la classe **PrintWriter** pour écrire dans le fichier csv les données et résultats du programme.
- Utilisation de boucles pour obtenir plus de résultats, itérations sur la taille du bloc (de 5 à 640 B, seulement pour établir le graphe au verso).
- Implémentation du design pattern décorateur pour les Benchmarkers, permettant de pouvoir créer de nouvelles sous-classes pour des benchmarks particuliers sans devoir modifier la classe principale.
- Classe **CsvWriter** pour l'écriture du fichier csv
- Classe **BenchmarkResult** pour pouvoir conserver les résultats d'un benchmark
- Classe **Experiment** comme contrôleur

	With Buffered Stream		Without Buffered Stream	
Block size (B)	**Reading (ms)**	**Writing (ms)**	**Reading (ms)**	**Writing (ms)**
500	17	18	28	64
50	16	15	259	423
5	32	77	2505	3277
0	61	187	11982	14977

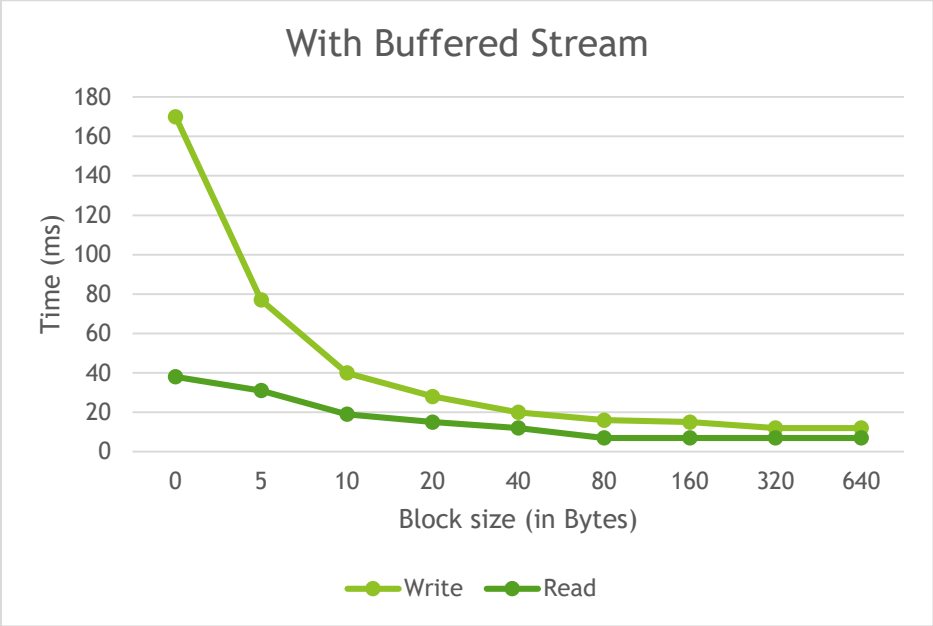


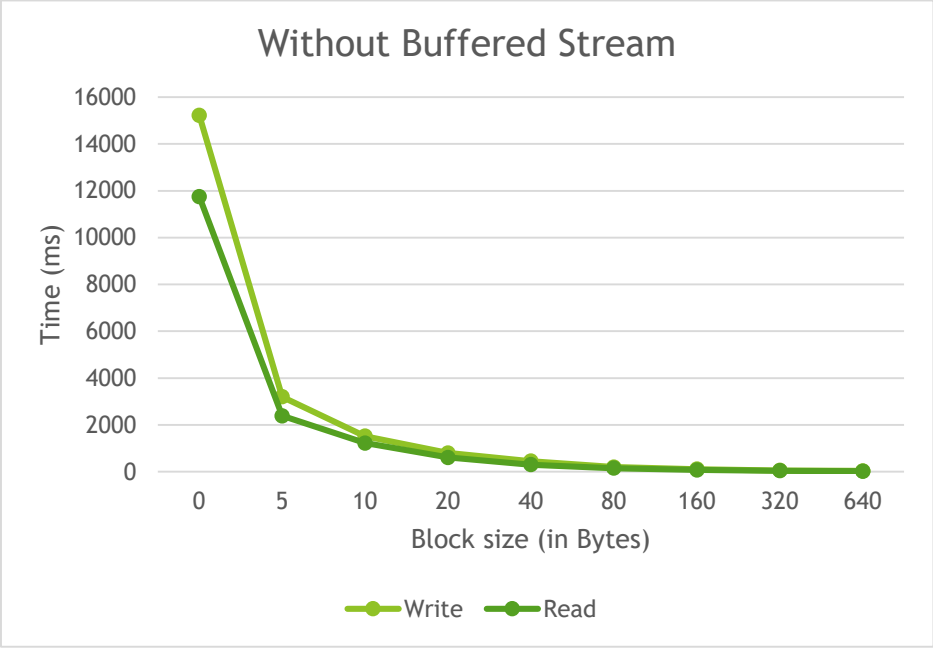






Résultats





Analyse

Nous pouvons constater que l'utilisation de flux avec tampons améliore les performances grandement :

Pour une écriture byte by byte, le temps passe de plus de 15 secondes à 170 millisecondes pour un fichier de 10MB

En revanche, pour une écriture par blocs de 640B, le temps est diminué par deux, ce qui est moins remarquable.

Nous pouvons en déduire que l'écriture et la lecture avec tampons se montre particulièrement utile pour une écriture par octet ou par petits blocs. Si une écriture sans tampon est utilisée, il faut alors écrire par blocs de grande taille (de préférence supérieure à 1KB) pour ne pas avoir des performances médiocres.