RES : Lab 01, Java IO

Toutes les données, les résultats et le code montré dans ce rapport sont disponible sur Github (https://github.com/BenjaminSchubert/JavaIOBenchmark)

1 Conditions de l'expérience

Dans cette expérience, nous allons comparer les performances de différentes tailles de tampons lors de l'écriture et la lecture sur un disque en Java afin de mesurer l'impact des-dits tampons sur les performances d'une application

Tous les tests ont été réalisés sur le système suivant :

- Processeur: Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz
- Vitesse d'écriture disque: 543 MB/s
- Vitesse de lecture disque: 7.3 GB/s
- Système d'exploitation: Linux
- Version du Kernel: 4.4.3-300.fc23.x86_64
- Version de java: openjdk version "1.8.0_72"
- Taille des fichiers : 100 MB
- Taille des blocks FS : 4096

2 Résultats

Les résultats sont présentés sur le graphe *Performance de l'I/O java*. Une vision plus précise des premiers points est donnée sur le graphe *Performance de l'I/O java : zoom sur les premières entrées*. Attention, l'axe Y utilise une échelle logarithmique. Chaque point représente l'écriture de 100 MB, avec sur l'axe X la taille des blocs utilisés et sur l'axe Y, le temps de complétion

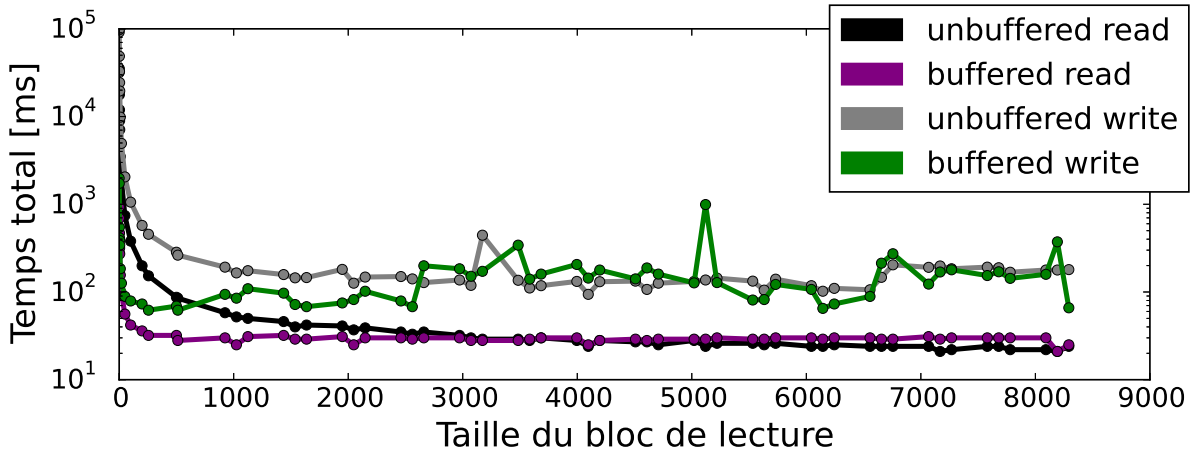

Figure 1: Performance de l'I/O java

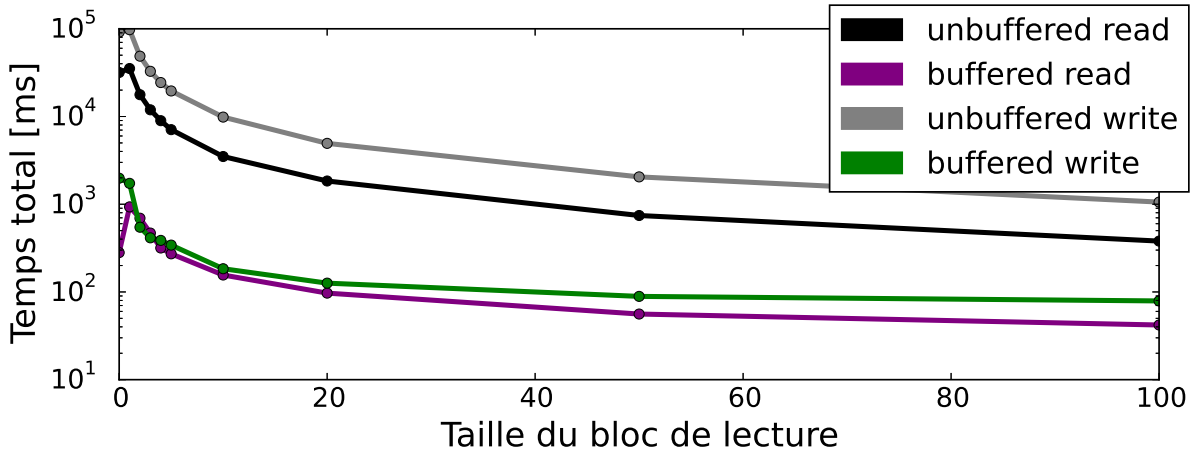

Figure 2: Performance de l'I/O java : zoom sur les premières entrées

3 Analyse

Sur le graphique *Performance de l'I/O java*, nous constatons que le temps de lecture diminue de manière proportionnelle à la taille des blocs de lecture jusqu'à une taille limite, qui correspond à la taille d'un blocs du système de fichier. Après ceci, les performances commencent à se dégrader.

Du côté de l'écriture, les performances avec un buffer, sont plus performantes pour de petites taille, bien que les grosses différentes dans les performances limitent les analyses pouvant être faites ici.

Nous pouvons aussi noter une amélioration locale des performances à chaque fois que la taille de lecture/écriture est un multiple ou diviseur de la taille de blocs du système de fichier.

Nous pouvons noter qu'à partir d'une certaine taille de bloc, soit la taille des blocs du système de fichier, la lecture avec un tampon devient légèrement moins efficace que celle sans tampon. Il serait donc bon dans les applications, lors de grosses lectures, de ne pas utiliser de tampon.

Nous pouvons aussi constater que les performances d'écritures sont beaucoup moins régulières que les performances en lecture, mais dans la tendance générale, nous pouvons dire que le tampon reste plus performant mais avec moins de régularité.

Pour conclure nous pouvons voir qu'il vaut mieux utiliser des tampons lorsque nous lisons et écrivons de petits blocs de données, et qu'à partir d'une certaine taille, le tampon n'ajoute plus grand bénéfice. Le point le plus important étant qu'il faut que la quantité de données lues ou écrites ou la taille du tampon soit un multiple ou un diviseur de la taille des blocs du système de fichiers afin d'observer les meilleures performances.

4 Mise en place de l'expérience

Pour réaliser cette expérience, nous avons créé un loggeur, *AbstractTestResultLogger*, spécialisé dans le résultat des tests que nous attendions. Celui-ci prend un *OutputStream* quelconque en paramètre et l'entoure avec un *BufferedOutputStream*, pour en améliorer les performances. Celui-ci est ensuite entouré avec un *OutputStreamWriter*, afin de pouvoir écrire facilement avec des *Strings*. Ce loggeur implémente aussi *AutoCloseable*, afin d'en faciliter la gestion, grâce aux nouvelles capacités implémentées en Java 7.

Nous avons ensuite spécialisé ce loggeur, pour qu'il enregistre ses données en CSV, dans la classe *CSVResultLogger*. Dans le programe principal, nous passons un *FileOutputStream* à ce loggeur, afin d'écrire ces données dans un fichier.

Nous avons ensuite remplacé la partie principale du programme qui ne faisait que quelques tests par une série de tests, afin d'avoir plus de points de mesure. Nous avons finalement mis à parti le nouveau *AutoCloseable* de Java 7 afin de réduire les risques potentiels avec l'ouverture de flux.