Réseau : Mesure des performances des entrées-sorties en Java **Labo 02**

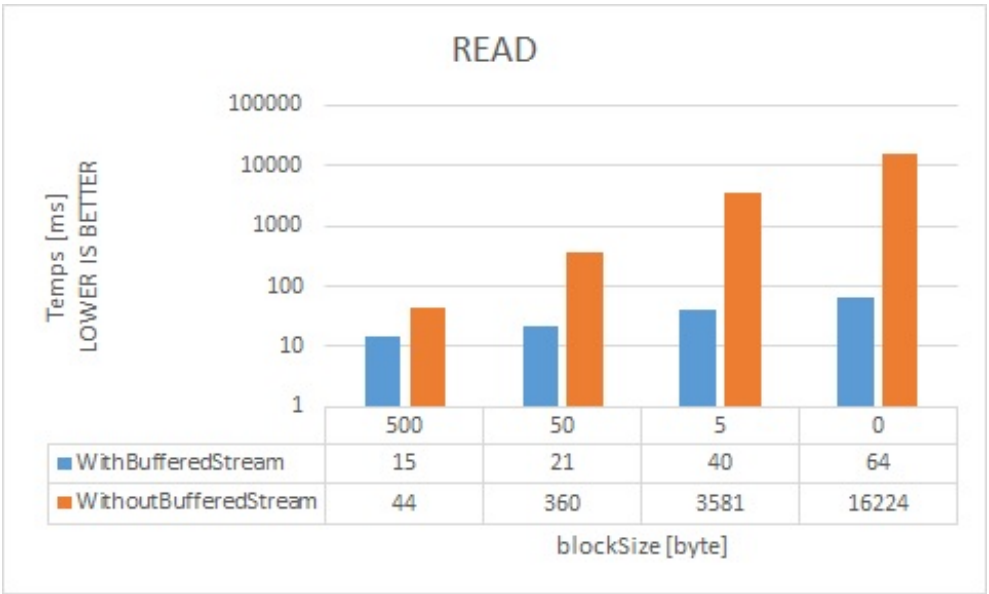
1 Conditions de l'expérience

Pour ce laboratoire, j'ai utilisé mon ordinateur portable, doté d'un processeur i7-3630QM @ 2.40 GHz. J'ai limité le nombre de programme tournant en parallèle pour avoir le maximum du CPU disponible. Cette ordinateur est aussi équipé d'un disque dur Hitachi 5400trs/min avec des tailles de file system de 4Ko.

Tous les tests sont effectués sur des fichiers de 10Mo

2 Mesures

Les résultats sont représentés sur les deux graphes ci-dessous. Il est important de noter que les échelles de temps sont des échelles logarithmiques.



3 Analyse des mesures

En se basant sur les deux précédents graphes, on peut constater que pour ce qui est de l'écriture ou la lecture, le temps croît de manière similaire inversement proportionnelle à la taille des blocs.

La partie intéressante vient donc dans les parties bufferisée. En effet, on peut constater que le temps continue à croître mais de manière bien plus lente. Ceci s'explique par le fonctionnement même d'un buffer.

Les flux bufferisés ont effectivement moins d'appel système respectivement de lecture ou d'écriture à faire et sont donc plus efficace. Cependant, on peut voir que cette différence à tendance à diminuer plus la taille du bloc est grand.

On peut donc en conclure que si l'on veut lire un très grand nombre de données, on a meilleur temps de lire avec des tailles de bloc très grand et de tout manière utiliser des flux bufferisés.

4 Modification

Ils nous a été demandé d'inscrire les données dans un fichier CSV. Je me suis donné comme ligne directrice de modifier le stricte minimum ainsi que de rajouter le moins de ligne possible tout en restant le plus réutilisable possible.

Pour cela, j'ai donc crée une classe CSV avec une fonction de log qui écrit directement dans le fichier avec le formatage CSV demandé. Le constructeur de cette classe prends un OutputStream pour ne pas simplement écrire dans un fichier mais dans n'importe quel type de flux de sortie.

J'ai ensuite appelé ma fonction de log à la fin des fonctions "produceTestData" et "consumeTestData". Le retour de la fonction "consumeDataFromStream" a du être modifié pour qu'elle puisse retourner le nombre de bytes lu afin de pouvoir l'inscrire dans le log.

À noter que j'ai préféré ne pas m'occuper des exceptions et ai décidé de simplement les throws.