Nous avons conclus que l’approche de regroupement de threads était celle la plus efficace généralement, car comme nous avons pu constater, elle est en moyenne plus efficace pour le travail de traitement que nous avions à faire que l’approche séquentiel et parallèle.

En premier lieu, lorsque nous comparons les données de vitesse de séquentiel avec ceux des deux autres on constate rapidement que cette approche ne devient plus viable dès que l’on nécessite une performance accrue. En effet, lorsque nous n’utilisons qu’un seul cœur pour effectuer un grand nombre de transformations, une grande partie des ressources disponibles de l’ordinateur sont simplement en attente et donc ne contribuent pas à augmenter la vitesse d’exécution des transformations. Ainsi, plus on augmente le nombre de fichier plus l’écart entre les solutions utilisant de la multiprogrammation ce fait ressentir. Cette approche n’est cependant pas à rejeter dans toutes les situations, car dans un cas où il serait possible de prendre plus de temps à effectuer une tâche sans altérer l’expérience de l’utilisateur, il serait judicieux de l’envisager puisque celle-ci ne prend pas beaucoup de ressource du processeur lui permettant ainsi de se concentrer sur d’autre tâche et de ne pas le surcharger.

En deuxième lieu, le

En conclusion, le regroupement de threads est donc la solution la plus efficace pour le travail