**Projet Big Data - Partie 2**

**Groupe 35 : Abderrazzak Mahraye - Noemane Chahid - Sophie Marimbordes**

Nous avons traité la partie infrastructure du projet en utilisant l’option Docker.

Le but du projet est de compter le nombre de mots d’un gros fichier texte (512 Mo) en utilisant la technique MapReduce. Le framerwork Spark est utilisé pour distribuer la tache sur 1 à 4 slaves qui sont des containers Docker. L’opération est coordonnée par un master, également un container Docker. A chaque container est associé un cœur de la machine ce qui permet de paralléliser l’exécution de la tâche.

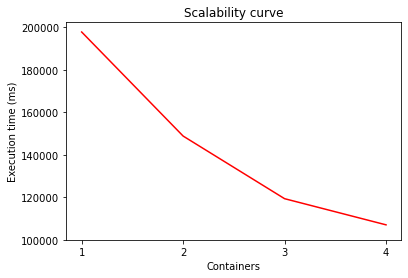
Nous observons effectivement une diminution du temps d’exécution de l’application WordCount lorsque le nombre de slaves augmente (cf. page 2).

1 seul container 🡺 197 855 ms soit 3.3 min

2 containers 🡺 148 856 ms soit 2.5 min

3 containers 🡺 119 362 ms soit 2 min

4 containers 🡺 107 045 ms soit 1.8 min

Courbe de scalabilité :

En exécutant le programme Count.java (exécution séquentielle), nous avons un temps d’exécution plus petit (11 390 ms soit environ 11 s) par rapport au MapReduce.

Si nous avions utilisé véritablement une architecture répartie, nous aurions obtenu des temps d’exécution plus petits avec l’utilisation des slaves. Dans notre cas, les containers (slaves) sont simulés sur notre PC via Docker, nous n’utilisons pas de vraies machines. Chaque slave ne possède pas des performances équivalentes à un node AWS.

