Le jeu de données utilisé pour ce TP a été publié par Google. Il représente 29 jours d'activité d'une machine à large échelle (un cluster de vers 12500 machines) de Google. Il contient des informations sur les jobs exécutés (chaque job mapreduce étant composé de plusieurs tâches) et des ressources utilisés par les tâches. Le jeu de données est composé de six parties, mais on n'en utilisera qu'une : task_events, disponible sur /raw_data/Google/task_events.

Dans ce répertoire, 1,6 gigaoctets de données sont distribuées dans plusieurs fichiers .csv. Nous vous recommandons de travailler sur un seul fichier d'abord pour essayer votre code, et passer au jeu de données entier plus tard.

Le fichier /raw_data/Google/schema.csv donne la description du jeu de données, avec ce que chaque colonne des fichiers .csv veut dire. Pour des détails sur la colonne << event type >>, allez à la documentation officielle de ce jeux de données, sur la page 6 :
https://drive.google.com/open?id=0B5g07T_gRDg9Z0lsSTEtTWtpOW8

Remarquez que l'identifiant d'une tâche (<< task index >>) n'est unique que parmi les tâches d'un même job. Par contre, les identifiants des jobs (<< job ID >>) sont uniques.

Question 1

Ce jeu de données contient quelques lignes avec moins de colonnes que montré dans schema.csv. Avant démarrer l'analyse avec ces données, filtrez ces lignes.
Combien de lignes avez-vous filtrez ? Parmi les lignes qui n'ont pas le bon nombre de colonnes, on a de tâches générées par combien d'utilisateurs différents ?

Question 2

Affichez toutes les valeurs possibles pour la priorité d'une tâche (<< priority >>).

Question 3

Combien de tâches différentes ont été exécutées ? Elles font partie de combien de jobs différents ?

Question 4

Quelle est la pourcentage de tâches qui ont eu des événements du type KILL ou EVICT ?

Question 5

Quelle est le nombre moyen de tâches par job ? Fournir également la médiane et les premier et troisième quadrants.
Pour le calcul de la médiane et des quadrants, vous pouvez utiliser la méthode zipWithIndex() des RDDs.

Question 6

Mesurez le temps dépassé par des différents appels à fonctions de votre code. Expliquez les résultats.

Identifiez des résultats intermédiaires utilisés par votre analyze et utilisez la fonction persist() pour les garder en cache. Mesurez l'impact sur la performance.