C’est 4v : volume(taille), valeur(pertinents), vélocité(vitesse), variété(sources)  
  
Architecture batch : traitement data volumineuse sur le long terme avec ajout des ordis(apache hadoop) (ex: traitement des données historique)  
Architecture streaming : traitement data en temps réel avec la memoire pour avoir des resultats plus rapide (apache flink)  
Architecture lamda : traitement hybride (apache spark = machine learning) = traitement en memoire, api polyvalent, traitement par lots et réel. Avantages : vitesse de traitement, facilité d'utilisation, traitement temps réel. Utilisation : analyse temps réel, traitement interactive, machine learning)  
solution :   
systeme de fichiers distribués(Apache hadoop: HDFS, mapreduce, yarn) = stocker les données de manière évolutive. Avantages : évolutive horizontale, tolérance aux pannes, adapté traitement par lots.  
les clusters de calculs(utilisant des frameworks Hadoop, Spark) = traitement de données en parallèle sur un ensemble de noeuds interconnectés  
  
APACHE HADOOP VS APACHE SPARK :  
Hadoop = batch / Spark = temps réel et batch  
Spark + rapide que Hadoop  
Spark offre API plus riche pour dev  
  
Apache Spark : RDD.   
Caractéristiques :   
Résilience -> Tolérance aux pannes c'est à dire ils peuvent être reconstruits en cas de défaillance d'un noeud du cluster.  
Immutabilité -> Une fois crée, on peut pas le modifié afin de garantir la cohérence et la prédictibilité des résultats   
Partitionnement -> Les données sont divisées en partitions qui sont traitées en parallèle.

