



Cours Big Data

Master Recherche en Informatique Décisionnelle et Vision Intelligente (MIDVI)

Batch Processing

20 Min



VS

Real-Time Processing

Less Than 1 Sec



Préparé par :
Pr. Noura AHERRAHROU

Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

Mise en contexte

- ❑ Le MapReduce est un modèle de calcul dont l'exécution se fait en **batch**.
- ❑ Malgré sa simplicité, le MapReduce n'est pas adapté à toutes les problématiques, plus précisément celles qui impliquent un traitement **interactif** ou **itératif**.
- ❑ Pour contourner ce problème, plusieurs acteurs sur le marché ont proposé des modèles de calcul alternatifs au MapReduce. Certains sont également en **batch** et permettent d'élargir le type de problématiques qui peuvent être traitées sur un cluster, tandis que d'autres sont **interactifs** et offrent des temps de réponse très faibles.

Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

Traitement par lot: Batch Processing

Pour comprendre le traitement Batch (sur disque), il faut connaître les principes fondamentaux du traitement des données dans un ordinateur. Par principes, nous entendons des lois qui ne changent pas, ni avec le temps, ni avec la technologie. Quatre principes régissent le traitement de données dans un ordinateur.

1er principe : tout traitement informatique des données implique toujours la participation de trois composants de l'ordinateur : le disque dur, la mémoire centrale et le microprocesseur.

2e principe : le microprocesseur traite toujours les données à partir de la RAM. Toujours !

3e principe : la mémoire RAM garde son contenu le temps d'une session.

4e principe : le disque dur a été conçu pour la persistance des données.

Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

Traitement par lot: Batch Processing

Dans la pratique, les données sont chargées par lots (ou par paquets) en mémoire et traitées au fur et à mesure par le microprocesseur ; c'est ce qu'on nomme batch processing ou traitement par lots.

Par son principe, le **batch processing** garantit au système la capacité de **traiter de très gros volumes de données**. **En contrepartie**, les traitements sont ralentis par les multiples **allers-retours**, parce qu'ils rendent le système inexploitable face aux problématiques nécessitant des réponses rapides (d'où son alternative, le **traitement in-memory**)



Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

Traitement par lot: Batch Processing

- ☐ Traitement hors-ligne
- ☐ Temps de réponse long : minutes, heures ...
- ☐ Les données doivent être disponibles sur disque avant le traitement
- ☐ Le traitement batch s'applique à de gros datasets que l'on peut éventuellement réunir Caractéristiques Volume et Variété des 5V
- ☐ Lecture/écriture séquentielle
- ☐ Temps de latence élevé
- ☐ Traditionnellement les données sont déplacées d'un noeud de stockage à un noeud de traitement
- ☐ Exécution simultanée de parties du traitement sur les différents noeuds, Ensuite regroupement des différents résultats

Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

Batch Processing:Mapreduce



- ❑ Hadoop Map-Reduce est un exemple de système utilisant le traitement par lot
- ❑ Le MapReduce représente un changement mental quant à la façon dont on a toujours écrit les algorithmes ; il consiste à découper le traitement d'un fichier de données en tâches indépendantes, en suivant deux phases : une phase **Map** et une phase **Reduce**. L'utilisateur spécifie une fonction de hachage Map, qui transforme les données d'entrée en paires clé/valeur, et une fonction de hachage Reduce, qui agrège toutes les valeurs associées à la même clé. Une phase intermédiaire appelée *Shuffle* trie les paires clé/valeur selon la clé. Ce genre de programme, écrit dans un style fonctionnel, est automatiquement parallélisé et exécuté dans un cluster.

Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

Mapreduce : des bénéfices et gains incontestables

- **La robustesse** : map reduce d'Hadoop n'est pas sensible à la panne d'un serveur du cluster. En d'autres termes, le framework Hadoop se comporte de la même façon même en cas de pannes de serveurs.
- **La flexibilité** : un des grands intérêts d'Hadoop est de pouvoir stocker et analyser aussi bien des données structurées que des données non structurées, et ceci de la même manière !
- **La rapidité** : la structure distribuée d'Hadoop HDFS (hadoop data file system) permet de localiser et de récupérer la donnée où qu'elle se trouve. Ainsi, Hadoop permet de traiter des requêtes en quelques minutes sur des téraoctets de données et sur des pétaoctets en quelques heures.



Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

Mapreduce des bénéfices et gains incontestables

- **La performance** : contrairement aux bases de données traditionnelles, Hadoop permet aux applications d'interroger et d'analyser des téraoctets de données très rapidement tout en garantissant les performances, en utilisant un système de serveurs très bon marché (cluster parallélisé).
- **Le coût de stockage** : Hadoop permet de stocker à moindre de coût ! Avec un tel framework, il devient alors possible de stocker et de conserver toutes les données, pour le plus grand bonheur de nos Data Scientist.



Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

Mapreduce : les limites



- ❑ Malgré tous ces avantages, Hadoop-Mapreduce est toujours en pleine mue pour être encore plus efficace. En effet, l'un des plus grands inconvénients d'Hadoop réside dans la **latence des traitements**. Le fait que toute requête utilise le map reduce (mapper les données et les réduire en exploitant l'ensemble des serveurs), il existe un temps de latence incompréhensible qui dépend aussi du nombre de serveurs sollicités.
- ❑ Le MapReduce fonctionne selon un modèle batch ; il n'est pas adapté aux trois types de problématiques suivantes : **les analyses interactives**, **les jobs itératifs** et **le streaming**.

Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--



Mahout et Hama

Apache Hama



- ❑ Le manque d'itérations imposé par le MapReduce complique son utilisation sur certaines problématiques.
- ❑ Pour exploiter les données, les entreprises utilisent traditionnellement les techniques provenant de disciplines telles que la statistique, le data mining, la recherche opérationnelle et l'intelligence artificielle. Ces dernières fournissent des algorithmes dits « d'apprentissage » qui, appliqués aux données, arrivent soit à en prédire les valeurs futures (c'est la prédiction), soit à les regrouper par indice de similarité (on parle de classification). On peut citer, entre autres, les algorithmes de régression, de clustering, de recommandation et d'exploration.

Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

Mahout et Hama



Apache Hama

Passer au numérique implique donc l'adoption d'Hadoop, mais également une transition des algorithmes vers le calcul parallèle. Or, souvenez-vous qu'un algorithme ne peut s'exécuter sur cluster Hadoop que s'il est parallélisable. Dès lors, faire transiter tous les algorithmes d'analyse de données revient à les paralléliser, ce qui est le but de **Mahout** et **Hama**, dont nous allons maintenant vous donner un aperçu.

Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

Mahout



- ❑ Mahout est un projet de la fondation Apache, qui a pour but de fournir des implémentations MapReduce des algorithmes d'apprentissage pour le traitement des données sur un cluster Hadoop.
- ❑ Concrètement, Mahout n'est pas un logiciel, mais une bibliothèque de classes Java pour les algorithmes d'apprentissage. Cette bibliothèque peut être utilisée par n'importe quel moteur de calcul tels que Hadoop, Spark ou n'importe quel moteur de calcul propriétaire peuvent s'appuyer sur les algorithmes fournis par Mahout pour permettre aux utilisateurs de réaliser des travaux de machines learning sur le cluster.



Apache Hama

- ❑ Hama est un moteur de calcul complémentaire qui renforce Hadoop dans les problématiques du calcul scientifique massivement parallèle.
- ❑ Hama est recommandé pour les travaux scientifiques (physique, chimie, apprentissage statistique). Il fournit, entre autres, les opérations de multiplication de matrices, d'équations linéaires, de calcul de graphes, de calcul de valeurs et vecteurs propres.

Traitement in memory: Streaming Processing

L'in-memory processing désigne le fait de traiter les données en mémoire. C'est le contraire du batch processing. Concrètement, cette approche consiste à charger tout le fichier de données en mémoire. De fait, les multiples opérations d'I/O sont supprimées et le processeur a un accès direct aux données en mémoire. Par conséquent, les traitements s'exécutent avec une latence très faible.

L'in-memory processing est le mode utilisé par défaut dans presque toutes les applications de statistiques et de data mining.



Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

Traitement in memory: Streaming Processing

L'inconvénient de ce mode de traitement, vous vous en doutez, c'est évidemment de limiter la taille de fichier imposée par la capacité de la RAM. Dans un contexte de Big data où les fichiers dépassent déjà la barre des téraoctets, il est tout simplement impossible de faire appel à cette approche. C'est un problème sérieux pour les analystes et les utilisateurs métier qui, avec l'accroissement du volume de données, perdent l'interactivité de leurs applications préférées et voient le temps de réponse à leurs requêtes augmenter sérieusement. Pour résoudre ce problème, deux solutions ont été développées.



Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

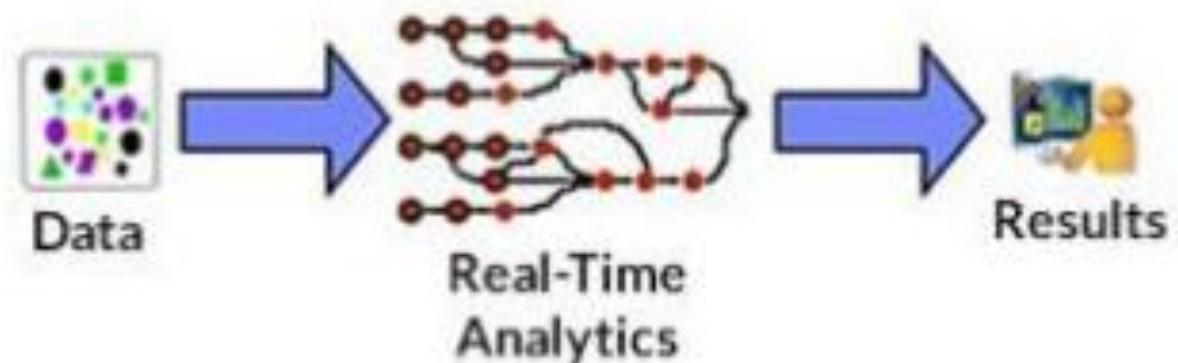
Traitement in memory : Streaming Processing

- ❑ La compression des données chargées en mémoire. C'est l'une des techniques utilisées pour continuer à faire de l'in-memory processing sur de très gros volumes de données. Elle consiste, à l'aide de techniques issues du traitement du signal, à réduire la taille du fichier sans altérer sa qualité, avant de le charger en mémoire pour traitement.
- ❑ L'ajout de mémoire RAM dans l'infrastructure informatique. Cette deuxième solution pour pallier à la limitation de la taille de la RAM consiste à ajouter des barrettes mémoire supplémentaires dans l'infrastructure informatique. Cette approche s'appuie sur la baisse des prix de la RAM, qui s'accompagne d'une amélioration de sa capacité.

Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

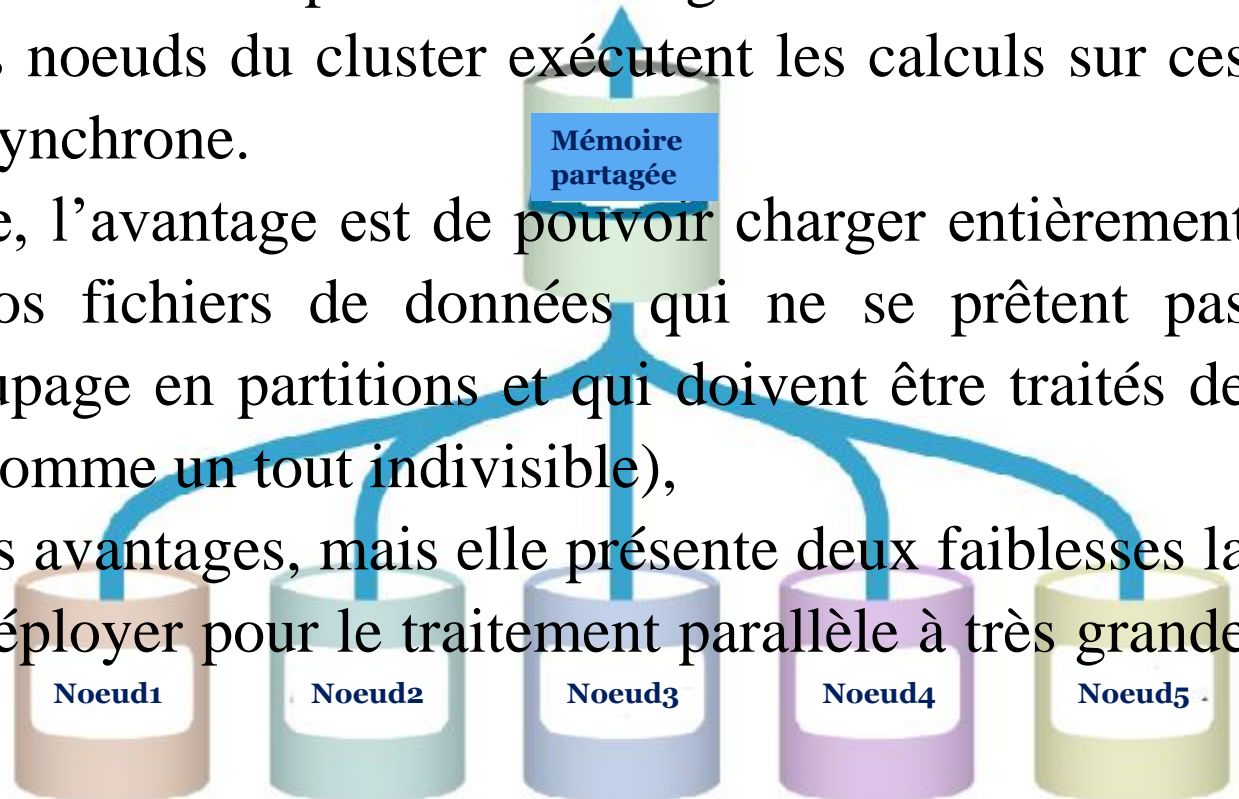
Traitement in memory : Streaming Processing

Il existe deux approches pour faire du traitement parallèle en in-memory. La première consiste à configurer le cluster selon une *architecture **shared-memory***, et la seconde à utiliser une approche ***shared-nothing*** dans laquelle les données sont découpées, distribuées entre les noeuds du cluster et chargées dans la mémoire individuelle de chaque noeud.



L'in-memory en architecture de cluster shared-memory

- ❑ Dans cette approche, le cluster est configuré de sorte que les noeuds partagent la même mémoire RAM : on parle d'architecture shared-memory. Les données sont complètement chargées dans la mémoire partagée, et tous les noeuds du cluster exécutent les calculs sur ces données de façon asynchrone.
- ❑ Dans cette approche, l'avantage est de pouvoir charger entièrement en mémoire de gros fichiers de données qui ne se prêtent pas facilement au découpage en partitions et qui doivent être traités de façon asynchrone (comme un tout indivisible),
- ❑ Cette approche a des avantages, mais elle présente deux faiblesses la rendant difficile à déployer pour le traitement parallèle à très grande échelle.



Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

L'in-memory en architecture de cluster shared-memory

- ❑ Lorsque plusieurs noeuds accèdent à la même mémoire, ils doivent utiliser un protocole de communication comme l'IPC9 pour la modification des données qu'ils manipulent. Bien que ce protocole offre de hautes performances dans l'accès à la mémoire partagée, il est quelquefois surchargé, ce qui dégrade rapidement la performance du cluster.
- ❑ Le fait que la mémoire soit partagée plafonne le nombre de noeuds qui peuvent y accéder, limitant ainsi le dimensionnement et la *scalabilité* du cluster. Dans la pratique, la mémoire devient difficile à partager à partir de 10 noeuds.

C'est pour ces deux raisons que l'approche in-memory basée sur l'architecture shared-nothing a été mise sur pied.

Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

L'in-memory en architecture de clusters shared-nothing

La deuxième approche de traitement parallèle en in-memory est similaire au batch parallèle, à la seule différence que les données sont chargées en mémoire sur chaque noeud. Concrètement, le cluster est ici configuré dans son mode normal – c'est-à-dire en shared-nothing – où chaque noeud possède sa propre mémoire, son propre disque dur et son propre processeur. Le fichier de données est découpé en partitions affectées aux différents noeuds. Chaque noeud charge sa partition en mémoire et exécute le traitement séquentiel qui a été défini par l'utilisateur.

Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

La nouveauté de l'année 2013 - finie la frontière entre streaming et batch

En novembre 2013, Cloudera annonce supporter via sa distribution le projet Spark. Spark est un framework d'Apache de type map/reduce (Streaming-processing) qui s'interface avec Hadoop HDFS et permet d'exploiter plus efficacement les serveurs surtout sur des requêtes itératives qui sollicitent les données à plusieurs reprises. Et pour le bien de tous, Spark permet aussi de simplifier les codes map/reduce. Pour les initiés, le fameux traitement « wordcount » se code en 3 lignes sur Spark !



Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

Spark : le moteur in-memory distribué d'Hadoop

Spark est le tout premier modèle de calcul distribué in-memory à avoir été développé. Il s'agit même du principal moteur de ce type disponible sur le marché et à la fondation Apache. Il a été conçu à la fois pour le **calcul interactif** et le **calcul itératif**. Il affiche des performances dix fois plus élevées qu'Hadoop sur les travaux itératifs (apprentissage automatique).



Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

Spark : le moteur in-memory distribué d'Hadoop

L'université de Berkeley a mené une expérience au cours de laquelle elle a comparé Spark et Hadoop sur la performance d'un job de régression logistique, un algorithme itératif de classification qui essaye de trouver une droite (un hyperplan, pour être plus formel) séparant au mieux deux ensembles de données. Pour ce faire, un fichier de 29 Go a été utilisé sur un cluster de 20 machines possédant 4 CPU chacune. Les résultats sont là : avec Hadoop, chaque itération prend 127 s, tandis qu'avec Spark, la première itération est de 174 s, mais les suivantes de 6 s seulement.



Introduction Au Big Data	L'écosystème Hadoop	Batch vs. Streaming Processing		
-----------------------------	------------------------	--------------------------------------	--	--

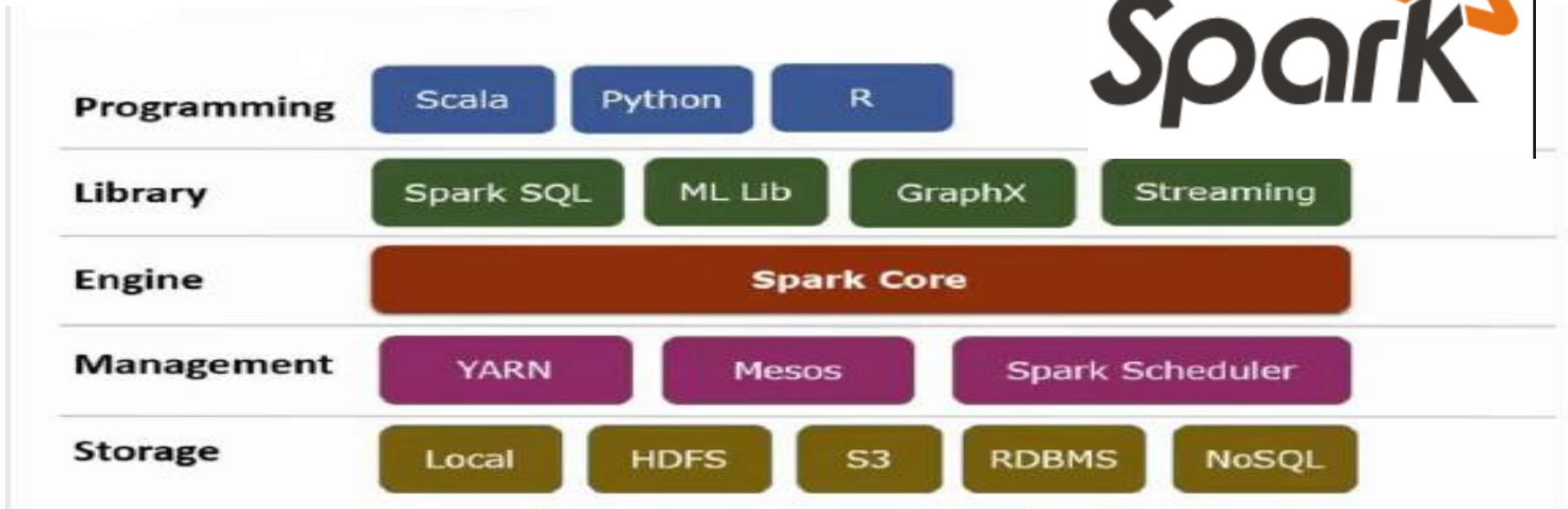
Spark : le moteur in-memory distribué d'Hadoop

Au cours d'une autre expérience, Spark a été utilisé pour charger 39 Go de fichiers Wikipédia en mémoire à travers un cluster de 15 machines, pour y effectuer ensuite des requêtes interactives. À la première requête, le traitement a duré 35 s. Les requêtes suivantes ont pris quant à elles de 0,5 à 1 s.

Ces expériences montrent que sur les travaux itératifs ou interactifs, Spark est de loin meilleur qu'Hadoop. Quelles sont les clés de sa performance ? C'est ce que nous allons voir.



Spark



- Framework efficace et générique
- Etend MapReduce à d'autres types de calculs
- Requêtes interactives
- Calculs en mémoire
- Plus rapide que Hadoop
- Requêtes SparSQL
- Plusieurs schedulers: standalone, YARN, Mesos
- Plusieurs filesystems dont HDFS
- Bibliothèques haut niveau
- Différents langages de programmation