**Hadoop**

**Plan :**

* Qu’est-ce que c’est ?
  + Licence
  + Contributeurs / Distributions
  + Destinataires
* Comment Hadoop résout le problème du Big Data ?
* Les indispensables d’Hadoop : HDFS et MapReduce
  + HDFS (Nodes)
  + MapReduce
  + YARN
* Ecosystème Hadoop
  + Outils de gestion et configuration de l’Application
  + Outils d’accès aux données

**Qu’est-ce que c’est ?**

Hadoop est un Framework Java open source permettant de développer des applications distribués et échelonables (scalables).

Il a été développé à l’origine par Google, et il est maintenant développé en tant que projet Open Source par Apache et de nombreux contributeurs comme Cloudera, Horton Works ou encore Facebook.

## Comment Hadoop résout le problème du Big Data ?

Hadoop repose sur des modèles de programmation simples pour assurer le traitement d’immenses volumes de données et les rendre disponibles sur des machines locales.

[Hadoop](https://data--flair-training.translate.goog/blogs/hadoop-tutorial/?_x_tr_sl=auto&_x_tr_tl=fr&_x_tr_hl=fr) se compose de 2 composants principaux dont je vais vous expliquer le fonctionnement.

**HDFS**

HDFS (Hadoop Distributed File System) qui est un modèle de stockage.

*NameNode*

Chaque disque dur est un nœud. Les disques peuvent être plus ou moins grand.

Les données sont répliquées trois fois afin d’assurer leurs validités. Elles peuvent être répliquées sur des nœuds différents et elles peuvent également être dupliquées plus ou moins de 3 fois.

L’avantage d’avoir un système distribué est que l’ajout de disque est facile. Les disques peuvent être peu cher, pas besoin de tout stocker sur un même disque ou de réadapter l’application à chaque fois que les disques sont trop chargés.

L’autre avantage est que les processus s’exécutent sur plusieurs disques. Les performances sont donc  
grandement augmentées.

*SecondaryNode, BackupNode et CheckpointNode*

Le NameNode enregistre les modifications des fichiers sous forme de Logs et le secondary NameNode les récupères pour ensuite modifier le NameNode. En cas d’arrêt du NameNode tous les logs reçus auront été traités par le NameNode, ce qui permet de récupérer rapidement un état stable.

Hadoop reste cependant un SPOF (Single Point Of Failure) et le secondary NameNode est surtout là pour répartir la charge sur le namenode et aussi l’aider en cas de crash. Il ne prend cependant pas le relai du NameNode en cas de panne mais lui permet de ne pas perdre de données et de lui faire récupérer son état rapidement. Il existe cependant des solutions pour remédier à ce SPOF :

D’autres nodes existent pour aider le NameNode : le checkpoint Node qui copie le NameNode en temps-réel et met à jour ses logs directement. Le Back-up Node est probablement le futur de ces Node et offre les mêmes fonctionnalités que le CheckPoint Node et permet au NameNode de lire dans ses données n’importe quand.

*EdgeNode*

Le Edge node est généralement utilisé en entrée de la base mais sert aussi à des traitements nécessitant beaucoup de calculs et aux modules d’administration. Cela permet que les NameNodes ne soient pas sollicités pour des tâches autres que la recherche dans ses données, et de cibler le matériel pour des traitements spécifiques.

**MapReduce**

MapReduce qui est un algorithme permettant d’effectuer des calculs parallèles et d’avoir des données fiables.

L’algorithme mapreduce est en général effectué sur chaque datanodes contenant les données afin de paralléliser les tâches.

MapReduce vérifie la validité des données en les triant et les comparant MapReduce s’exécute sur chaque node pour effectuer les tâches en parallèle.

MapReduce permet d’organiser les données à l’enregistrement, et éventuellement de les trier.

**YARN**

YARN qui est gestionnaire des ressources, responsable de l'allocation des ressources et de la planification des travaux.

**Ecosystème Hadoop**

L’écosystème Hadoop est gigantesque et de nombreux outils sont utiles à des solutions spécifiques. Ceux présentés dans ce schéma sont les principaux.

**HBase**

C’est un SGBD non R (BDD NoSQL) qui s’installe généralement sur le système de fichier HDFS et dispose de MapReduce.

Particularités :

Open Source

HDFS

Java

Tolérant aux pannes grâce à HDFS

Requêtage rapide

Les tables de cette base de données peuvent servir d’entrée pour les tâches MapReduce dans l’écosystème Hadoop, et peuvent aussi servir de sortie après traitement des données par MapReduce

**Outils de gestion et configuration de l’application**

**Outils d’accès aux données**

**Concurrents de Hadoop**

Il existe des solutions NoSql différentes d’Hadoop qui sont fiables et très efficaces. Notamment Cassandra (qui peut s’intégrer à Hadoop) et qui est similaire mais orienté Ecriture plus que lecture. C’est le plus gros concurrent de HBase.