Définition de Map Reduce:

– MapReduce, un modèle de programmation qui fournit un cadre pour **automatiser le calcul parallèle sur des données massives**. (Openclassroom)

– **MapReduce** est un patron d'architecture de développement informatique, inventé par Google1, dans lequel sont effectués des calculs parallèles, et souvent distribués, de données potentiellement très volumineuses, typiquement supérieures en taille à 1 téraoctet. (wikipedia)
Utilisé sur Amazon.com ou Facebook et maintenant de plus en plus dans le Cloud computing (à vérifier)

– Est implémenté dans Hadoop (de Apache Software Foundation).

– "*MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model, as shown in the paper.*" – Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: Simplified data processing on large clusters." (2004). Cité 13492 fois

Modèle de Programmation Map Reduce:

Extrait de l'article:

" *The computation takes a set of input key/value pairs, and produces a set of output key/value pairs. The user of the MapReduce library expresses the computation as two functions: Map and Reduce.*

Map, written by the user, takes an input pair and produces a set of intermediate key/value pairs. The MapReduce library groups together all intermediate values associated with the same intermediate key I and passes them to the Reduce function.

The Reduce function, also written by the user, accepts an intermediate key I and a set of values for that key. It merges together these values to form a possibly smaller set of values. Typically just zero or one output value is produced per Reduce invocation. The intermediate values are supplied to the user's reduce function via an iterator. This allows us to handle lists of values that are too large to fit in memory." page 2

En d'autres termes: (d'après open classroom)

C'est **Divisez pour régner** et ses trois étapes consistant, pour un problème initial donné, à:

1. **Diviser** : découper le problème initial en sous-problèmes;
2. **Régner** : résoudre les sous-problèmes indépendamment soit de manière récursive, soit directement s'ils sont de petite taille;

3.**Combiner** : construire la solution du problème initial en combinant les solutions des différents sous-problémes.

MapReduce c'est **Divisez pour distribuer pour régner** en ce sens que la stratégie mise en place pour exécuter un calcul sur des données massives consiste à découper les données en sous-ensembles de plus petite taille, que nous appelerons des **lots** ou des **fragments** dans la suite, et à affecter chaque lot à une machine du cluster permettant ainsi leur traitement en parallèle. Il suffira ensuite d'agréger l'ensemble des résultats intermédiaires obtenus pour chaque lot pour construire le résultat final.

MapReduce c’est deux fonctions:

Map: Consiste à appliquer une même fonction à tous les éléments de la liste;

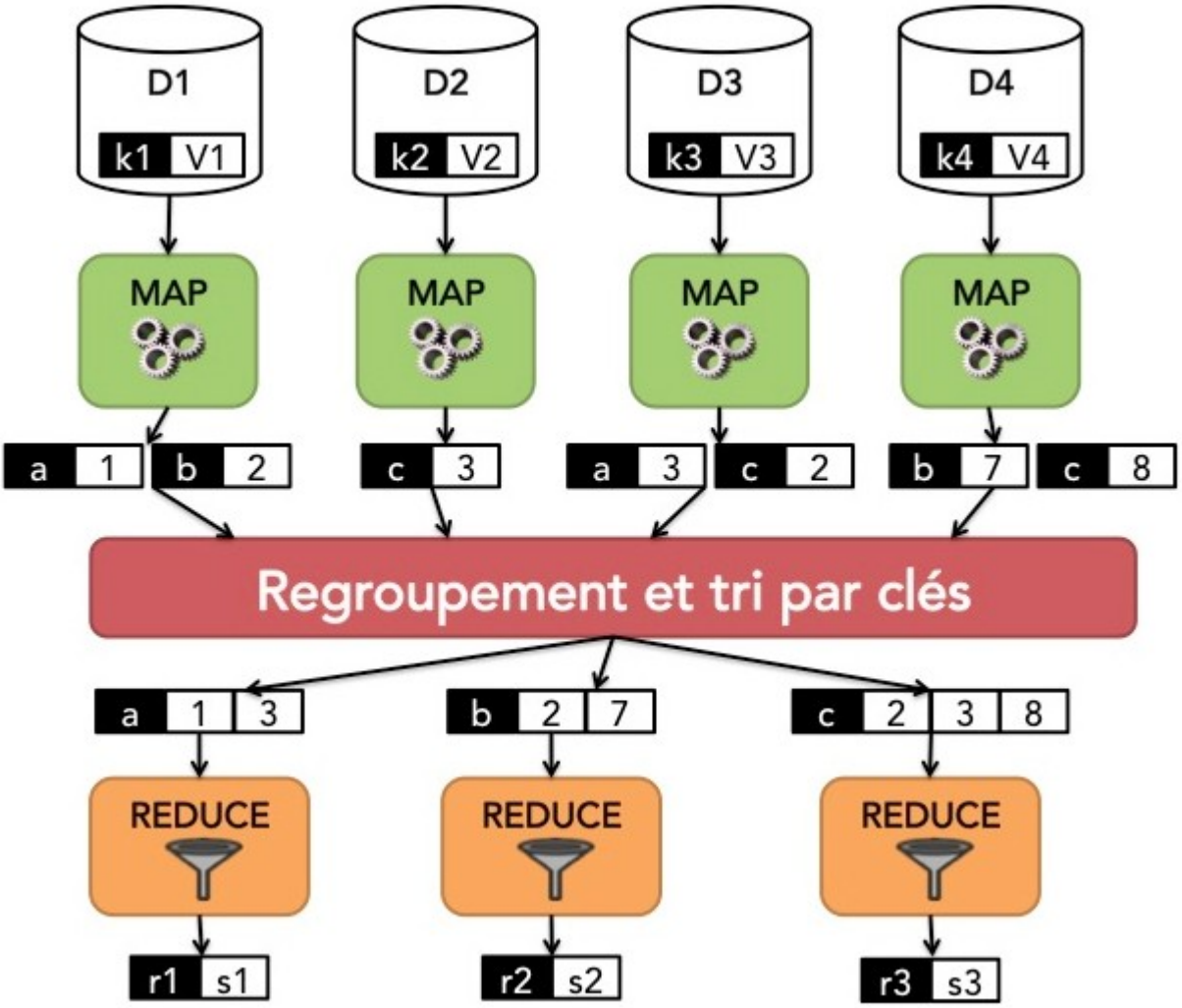
map(f)[x0, ..., xn] = [f(x0, ...,f(xn)]

*map(*4)[2, 3, 6] = [8, 12, 24]*

Reduce: applique une fonction récursivement à une liste et retourne un seul résultat;

reduce(f)[x0, ..., xn] = f(x0, f(x1, f(x2, ...)))

reduce(+)[2, 3, 6] = (2 + (3 + 6)) = 11



Le dernier ingrédient, fondamental pour l'automatisation de la parallélisation, est que **l'ensemble des données est représentée sous la forme de paires** `(clé, valeur)`, à l'instar des tables d'association.

Nous avons maintenant tous les ingrédients pour expliquer le fonctionnement général de MapReduce.

1. L'ensemble des données à traiter est découpé en plusieurs lots ou sous-ensembles.
2. Dans une première étape, l'étape MAP, l'opération `map`, spécifiée pour notre problème, est appliquée à chaque lot. Cette opération transforme la paire `(clé, valeur)` représentant le lot en une liste de nouvelles paires `(clé, valeur)` constituant ainsi des résultats intermédiaires du traitement à effectuer sur les données complètes.
3. Avant d'être envoyés à l'étape REDUCE, les résultats intermédiaires sont regroupés et triés par clé. C'est l'étape de `SHUFFLE and SORT`.

4. Enfin, l'étape REDUCE consiste à appliquer l'opération reduce, spécifiée pour notre problème, à chaque clé. Elle agrège tous les résultats intermédiaires associés à une même clé et renvoie donc pour chaque clé une valeur unique.

– Hello world: Fréquence des mots dans le texte.
– Multiplication matrice - vecteur
– Produit matriciel
– Inverse de matrices.