

ESD

Traitement de données massives

Programme

- Programmation fonctionnelle
 - Présentation
 - Mise en oeuvre avec Scala
- Hadoop et son écosystème
 - Hadoop
 - Système de données distribué
 - Extraction des données (Pig, Talend...)
 - Batch processing
 - SQL processing (Hive, Impala...)
- Spark
 - Concept
 - MLlib
 - Streaming

Programmation fonctionnelle

Présentation

- **Immutabilité**
- **Fonctions pures et transparence des références**
- **Fonctions d'ordre supérieur**
- **Recursion**
- **Eager vs Lazy**
- **Currying**
- **Map**
- **Reduce**
- **Monades**

Programmation fonctionnelle

Immutabilité

- L'immutabilité implique qu'une variable qui a été assignée ne peut l'être à nouveau.
- Le contenu de la variable ne peut être changé non plus.

Programmation fonctionnelle

Fonctions pures

- Une fonction pure est une fonction sans effet de bord.
- Chaque appel à cette fonction produira le même effet.
- Les appels étant idempotents, l'appel à ces fonctions peut être “lazy”.
- Simplifie les opérations suivantes:
 - Les tests, pas d'effets de bords, ni d'IO à gérer.
 - La combinaison des fonctions
 - La parallélisation
- La transparence des références est la garantie que pour une liste de paramètres donnés, le résultat sera toujours le même.
- La combinaison avec l'immuabilité permet la memoization (utilisation de la mémoire pour éviter du calcul cpu).

Programmation fonctionnelle

Fonctions d'ordre supérieur

- Une fonction d'ordre supérieur est une fonction qui retourne une autre fonction comme résultat.
- Il s'agit également d'une fonction qui peut prendre une autre fonction comme résultat.

Programmation fonctionnelle

Récursion

- **L'itération en programmation fonctionnelle n'existe pas.**
- **Pour parcourir un ensemble de valeurs, il faudra utiliser le mécanisme de récursion.**
- **Cela équivaut à appeler une fonction sur le résultat de la même fonction sur un sous ensemble des valeurs d'origine.**

Programmation fonctionnelle

Eager vs Lazy

- Le résultat d'exécution d'une fonction peut être “eager” soit calculé dès que l'appel à la fonction est déclaré.
- Ou “lazy”, l'exécution de la fonction n'aura lieu que si le résultat de celle-ci est utilisé dans la suite du programme.
- Cela permet d'ignorer les calculs inutiles ou de repousser l'exécution d'un traitement seulement lorsque celui-ci est réellement utile.

Programmation fonctionnelle

Currying

- Le currying est la transformation d'une fonction qui prend plusieurs paramètres en une séquence de fonctions qui prennent chacune un paramètre.
- Cela permet l'application de fonctions partielles.

Programmation fonctionnelle

Map

- La fonction map permet d'appliquer une transformation à l'ensemble d'une collection.
- Cela se traduit comme l'application d'une fonction unitairement à l'ensemble des éléments d'une collection.

Programmation fonctionnelle

Reduce

- Une opération de reduce permet de réduire la dimension d'un ensemble de valeurs.
- Cela veut dire que l'on applique une fonction permettant de combiner les éléments d'une collection entre eux.
- Le reduce est une opération d'agrégation entre plusieurs éléments similaires.
- Un reduce plus général est une agrégation devant définir un élément initial, la fonction à appliquer pour combiner un élément de la collection avec cet élément initial et la fonction permettant d'agréger plusieurs éléments similaires.

Programmation fonctionnelle

Monade

- Une monade est juste un monoïde dans la catégorie des endofuntors.

Programmation fonctionnelle

Question?



Programmation fonctionnelle

Monade

- Une monade est juste un monoïde dans la catégorie des endofoncteurs.
- Un endofoncteur est une fonction qui permet de passer d'un espace à lui même.
- Un monoïde est un ensemble muni d'une loi de composition interne associative et d'un élément neutre.
- En programmation fonctionnelle, une structure qui représente une séquence d'opérations séquentielles.
- Sert à calculer un état.
- Peut être simplifié comme une boîte encapsulant une valeur.

Programmation fonctionnelle

Mise en oeuvre avec Scala

- <https://www.scala-exercises.org>
- Notebook Zeppelin
- IntelliJ/Eclipse