

Spark Mise en oeuvre et programmation

Durée: 3 jours

1660 €

20 au 22 mars

26 au 28 juin

27 au 29 septembre

22 au 24 novembre

Public:

Chefs de projet, data scientists, développeurs.

Objectifs:

Savoir mettre en oeuvre Spark pour optimiser des calculs.

Connaissances préalables nécessaires:

Connaissance de Java ou Python, des bases Hadoop, et notions de calculs statistiques

Programme:

- Introduction** : Présentation Spark, origine du projet, apports, principe de fonctionnement
Langages supportés.
- Premiers pas** : Utilisation du shell Spark avec Scala ou Python
Gestion du cache
- Règles de développement** : Mise en pratique en Java et Python
Notion de contexte Spark
Différentes méthodes de création des RDD:
depuis un fichier texte, un stockage externe.
Manipulations sur les RDD (Resilient Distributed Dataset)
Fonctions, gestion de la persistance.
- Cluster** : Différents cluster managers : Spark en autonome, Mesos, Yarn, Amazon EC2
Architecture : SparkContext, Cluster Manager, Executor sur chaque noeud.
Définitions : Driver program, Cluster manager, deploy mode, Executor, Task, Job
Mise en oeuvre avec Spark et Amazon EC2
Soumission de jobs, supervision depuis l'interface web
- Intégration hadoop** : Travaux pratiques avec YARN
Création et exploitation d'un cluster Spark/YARN.
- Support Cassandra** : Description rapide de l'architecture Cassandra. Mise en oeuvre depuis Spark.
Exécution de travaux Spark s'appuyant sur une grappe Cassandra.

Spark Mise en oeuvre et programmation

- Spark SQL** : Objectifs : traitement de données structurées.,
Optimisation des requêtes.
Mise en oeuvre de Spark SQL.
Comptabilité Hive
Travaux pratiques:
en ligne de commande avec Spark SQL,
avec un pilote JDBC.
L'API Dataset :
disponible avec Scala ou Java.
Collections de données distribuées.
Exemples en Java.
- Streaming** : Objectifs , principe de fonctionnement : stream processing.
Source de données : HDFS, Flume, Kafka, ...
Notion de StreamingContexte, DStreams, démonstrations
Travaux pratiques : traitement de flux DStreams en Java.
- Mlib** : Fonctionnalités : Machine Learning avec Sparck,
algorithmes standards,
gestion de la persistance,
statistiques.
Support de RDD.
Mise en oeuvre avec les DataFrames.
- GraphX** : Fourniture d'algorithmes, d'opérateurs simples
pour des calcul statistiques sur les graphes
Travaux pratiques :
exemples d'opérations sur les graphes.