

## Formation Apache Spark

<b>Durée :</b>	4 jours
<b>Public :</b>	Développeurs, architectes système et responsables techniques qui veulent déployer des solutions Spark dans leur entreprise
<b>Pré-requis :</b>	Maîtrise de la programmation orientée objet en Java ou en C#
<b>Objectifs :</b>	- Développer des applications avec Spark - Utiliser les bibliothèques pour SQL, les flux de données et l'apprentissage automatique - Retranscrire des difficultés rencontrées sur le terrain dans des algorithmes parallèles - Développer des applications métier qui s'intègrent à Spark
<b>Sanction :</b>	Attestation de fin de stage mentionnant le résultat des acquis
<b>Taux de retour à l'emploi:</b>	Aucune donnée disponible
<b>Référence:</b>	BUS100299-F
<b>Note de satisfaction des participants:</b>	Pas de données disponibles

### Introduction

Définition du Big Data et des calculs  
À quoi sert Spark  
Quels sont les avantages de Spark

### Applications évolutives

Identifier les limites de performances des CPU modernes  
Développer les modèles de traitement en parallèle traditionnels

### Créer des algorithmes parallèles

Utiliser la programmation fonctionnelle pour l'exécution des programmes en parallèles  
Retranscrire des difficultés rencontrées sur le terrain dans des algorithmes parallèles

### Structures de données parallèles

Répartir les données dans le cluster avec les RDD (Resilient Distributed Datasets) et les DataFrames  
Répartir l'exécution des tâches entre plusieurs nœuds  
Lancer les applications avec le modèle d'exécution de Spark

### Structure des clusters Spark

Créer des clusters résilients et résistants aux pannes  
Mettre en place un système de stockage distribué évolutif

### **Gestion du cluster**

Surveillance et administration des applications Spark  
Afficher les plans d'exécution et les résultats

### **Choisir l'environnement de développement**

Réaliser une analyse exploratoire avec le shell Spark  
Créer des applications Spark autonomes

### **Utiliser les API Spark**

Programmation avec Scala et d'autres langages compatibles  
Créer des applications avec les API de base  
Enrichir les applications avec les bibliothèques intégrées

### **Interroger des données structurées**

Traiter les requêtes avec les DataFrames et le code SQL embarqué  
Développer SQL avec les fonctions définies par l'utilisateur (UDF)  
Utiliser les ensembles de données aux formats JSON et Parquet

### **Intégration à des systèmes externes**

Connexion aux bases de données avec JDBC  
Lancer des requêtes Hive sur des applications externes

### **Qu'appelle-t-on flux de données ?**

Utiliser des fenêtres glissantes  
Déterminer l'état d'un flux de données continu  
Traiter des flux de données simultanés  
Améliorer les performances et la fiabilité

### **Traiter les flux des sources de données**

Traiter les flux des sources intégrées (fichiers journaux, sockets Twitter, Kinesis, Kafka)  
Développer des récepteurs personnalisés  
Traiter les données avec l'API Streaming et Spark SQL

### **Classer les observations**

Prévoir les résultats avec l'apprentissage supervisé  
Créer un élément de classification pour l'arbre de décision

### **Identifier les schémas récurrents**

Regrouper les données avec l'apprentissage non supervisé  
Créer un cluster avec la méthode k-means

## **Développer des applications métier avec Spark**

Mise à disposition de Spark via un service Web RESTful  
Générer des tableaux de bord avec Spark

## **Utiliser Spark sous forme de service**

Service cloud vs. sur site  
Choisir un fournisseur de services (AWS, Azure, Databricks, etc.)

Développer Spark pour les clusters de grande taille  
Améliorer la sécurité des clusters multifournisseurs  
Suivi du développement continu de produits Spark sur le marché  
Projet Tungsten : repousser les performances à la limite des capacités des équipements modernes  
Utiliser les projets développés avec Spark  
Revoir l'architecture de Spark pour les plateformes mobiles