

Cycle certifiant Développeur BigData

CB070

Durée: 23 jours

11 720 €

Public :

Chefs de projet, développeurs

Objectifs :

- Savoir développer des applications bigData et mettre les oeuvre les outils adaptés aux différents types de données (Cassandra, spark, Neo4j, ElasticStack).
- Comprendre le fonctionnement des principaux outils de requête et de traitement des données.
- Acquérir les bonnes pratiques de distribution de messages, savoir configurer kafka pour intégrer les données de différents formats et de sources différentes.
- Comprendre les connexions existantes entre les mondes relationnels et NoSQL en environnement Big Data.
- Savoir mettre en oeuvre Hive et Pig, Impala, les Spark Dataframes.

Connaissances préalables nécessaires :

Connaissance des bases des systèmes d'information, et notions de calculs statistiques et d'un langage de programmation

Programme :

Comprendre les principaux concepts du Big Data ainsi que l'écosystème technologique d'un projet Big Data

L'essentiel du BigData : calcul distribué, données non structurées. Besoins fonctionnels et caractéristiques techniques des projets. La valorisation des données. Le positionnement respectif des technologies de cloud, BigData et noSQL, et les liens, implications.

Concepts clés : ETL, Extract Transform Load, CAP, 3V, 4V, données non structurées, prédictif, Machine Learning.

L'écosystème du BigData : les acteurs, les produits, état de l'art. Cycle de vie des projets BigData.

Atelier : Démonstration d'un prédiction Machine Learning avec Dataiku DSS

Savoir analyser les difficultés propres à un projet Big Data

Rôle de la DSI dans la démarche BigData. Gouvernance des données: importance de la qualité des données, fiabilité, durée de validité, sécurité des données

Emergence de nouveaux métiers : Data-scientists, Data labs, Hadoop scientists, CDO, ...

Intégration avec les outils statistiques présents et les outils BigData futurs.



Phirio

Déterminer la nature des données manipulées

Les différents modes et formats de stockage.
Les types de bases de données : clé/valeur, document, colonne, graphe. Besoin de distribution. Définition de la notion d'élasticité. Principe du stockage réparti.
Données structurées et non structurées, documents, images, fichiers XML, JSON, CSV, ...

Atelier : démonstrations avec une base MongoDB et une base Cassandra sur des données de différents types.

Appréhender les éléments de sécurité, d'éthique et les enjeux juridiques

Les risques et points à sécuriser dans un système distribué.
Aspects législatifs et éthiques: sur le stockage, la conservation de données, ..., sur les traitements, la commercialisation des données, des résultats

Atelier : mise en évidence des problèmes liés à la réplication inter-régions et concernant les aspects juridiques des données : droits d'exploitation, propriété intellectuelle, ...

Etude des failles de sécurité sur une infrastructure Hadoop.

Exploiter les architectures Big Data

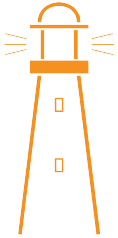
Les objectifs de la supervision, les techniques disponibles. La supervision d'une ferme BigData.
Objets supervisés. Les services et ressources. Protocoles d'accès. Exporteurs distribués de données.
Définition des ressources à surveiller. Journaux et métriques.
Application aux fermes BigData : Hadoop, Cassandra, HBase, MongoDB
Besoin de base de données avec agents distribués, de stockage temporel (timeseriesDB)
Produits : Prometheus, Graphite, Elasticsearch.
Présentation, architectures.
Les sur-couches : Kibana, Grafana.

Atelier : mise en oeuvre de prometheus pour la supervision d'une ferme Cassandra sur une infrastructure distribuée multi-noeuds.

Mettre en place des socles techniques complets pour des projets Big Data.

Etude des différents composants d'une infrastructure BigData :
Datalake : collecte des différents types de données
Stockage distribué : réplication, sharding, gossip, hachage,
Principe du schemaless, schéma de stockage, clé de distribution, clé de hachage
Systèmes de fichiers distribués : GFS, HDFS, Ceph. Les bases de données : Redis, Cassandra, DynamoDB, Accumulo, HBase, MongoDB, BigTable, Neo4j, ...
Calcul et restitution : Apport des outils de calculs statistiques
Langages adaptés aux statistiques, liens avec les outils BigData.
Outils de calcul et visualisation : R, SAS, Spark, Tableau, QlikView, ...
Caractéristiques et points forts des différentes solutions.

Atelier : mise en oeuvre du sharding avec une base de données MongoDB sur une infrastructure distribuée



Phirio

Elastic Stack

Elastic Stack

ElasticStack

Présentation, fonctionnalités, licence
Positionnement Elasticsearch et les produits complémentaires : Kibana, X-Pack, Logstash, Beats
Principe : base technique Lucene et apports d'ElasticSearch
Définitions et techniques d'indexation

Installation de base

Prérequis techniques.
Installation avec les RPM

Outils d'interrogation

Communication en RESTful avec le cluster
Interface http DevTools, travaux pratiques, démonstration

Traitement des données

Structure des données. stockage, indexation
Format des données.
Conversion au format JSON des données à traiter.
Interrogations avec Search Lite et avec Query DSL (domain-specific language)
Notion de 'filtre' pour affiner des requêtes.

Autres composants

Démonstrations de Logstash, Kibana et Beats
Intégration

Positionnement

Positionnement d'Elasticsearch et des produits complémentaires : Kibana, Logstash, Beats, X-Pack
Présentation des concepts clés d'ElasticSearch

Atelier : comparaison d'ElasticSearch avec d'autres moteurs d'indexation
en fonction du cas d'utilisation



Phirio

Identifier les enjeux et les cas d'utilisation d'un moteur de recherche

Présentation de la pile elastic.

Cas d'usage classiques : analyse de logs et sécurité, analyse de métriques, recherches web, ...

Analyse et visualisation de données

Principes de base de l'analyse de texte, recherche dans des données structurées, recherche full text,

Atelier : choix d'un outil d'indexation en fonction des contraintes d'utilisation

Appréhender le fonctionnement d'ElasticSearch

Principe : base technique Lucene et apports d'ElasticSearch. Fonctionnement distribué

Concepts clés : index, types, documents, noeuds, clusters, shards et replica

Notions de datatypes et mappings

Opérations CRUD : exemples d'opérations basiques, création d'index et mappings

Atelier : Installation d'une pile Elastic et création d'un index

Savoir installer et configurer ElasticSearch

Prérequis techniques. Mode d'installation en grappe distribuée.

API RESTful en HTTP . Requêtes avec Search Lite et avec Query DSL (domain-specific language)

Atelier : premiers pas dans la console DevTools de Kibana.

Indexer des volumes importants de données

Format et stockage des données.

Conversion au format JSON des données à traiter.

Structure des données. Stockage, indexation. Terminologie Elasticsearch : notions de document,index.

Métadonnées : _index, _ID

Choix de l'identifiant par l'application avec l'API index, ou génération automatique d'un identifiant, ...

Indexation inversée. Outils d'interrogation.

Atelier : exemples de requêtes simples et plus complexes : recherche de «phrases», extraction de plusieurs documents, ...

Notion de pertinence du résultat : «score»

Utilisation de 'filtre' pour affiner des requêtes.

Atelier : Mise en oeuvre d'une agrégation de résultats.

Ecriture de requêtes complexes.

Notions d'agrégations,

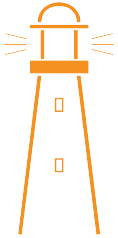
Atelier : préparation des données, agrégation de mesures, bucket aggregation.

Gestion des accès concurrents

Utilisation du numéro de version.

Gestion par l'application : différentes méthodes selon les contraintes fonctionnelles.

Utilisation d'un numéro de version externe.



Phirio

Comprendre comment administrer le système et le surveiller afin de garantir sa disponibilité

Flux logstash et présentation Kibana
Traitement de logs avec logstash
Introduction à beats, installation et configuration
Supervision du système avec MetricBeat et FileBeat

Atelier : recherche, visualisation, création de tableaux de bord et graphiques à partir des données fournies par Elasticsearch

Python pour l'analyse de données

Python pour l'analyse de données

Positionnement Python

Les valeurs d'observation, et les variables cibles.
Ingénierie des variables.
Analyses statistiques,
Classification des données, rapprochements,
Production de recommandations. Evolutions des outils statistiques classiques vers l'apprentissage automatique.

Atelier : exercices sur les outils statistiques de base

Savoir utiliser les principaux outils de traitement et d'analyse de données pour Python

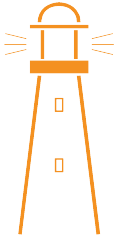
Besoins des data-scientists : calculs, analyse d'images, machine learning, interface avec les bases de données
Apports de python : grande variété d'outils, expertise dans le domaine du calcul scientifique
Présentation des outils d'apprentissage Python : scikit-learn, pybrain, TensorFlow/keras, mxnet, caffe

Atelier : mise en oeuvre de scikit-learn et génération de jeux de données.

Être capable d'extraire des données d'un fichier

Pandas : manipulation de tables de données. Notion de dataframe.
Manipulation de données relationnelles
Tableaux avec Pandas: indexation, opérations, algèbre relationnelle
Stockage dans des fichiers: CSV, JSon

Atelier : construction d'ETL de base entre json et csv



Phirio

Savoir appliquer les pratiques optimales en matière de nettoyage et de préparation des données avant l'analyse

Encodeurs
Filtres et ETL
Gouvernance des données. Qualité des données.
Transformation de l'information en donnée. Qualification et enrichissement.
Sécurisation et étanchéité des lacs de données.
Flux de données et organisation dans l'entreprise. De la donnée maître à la donnée de travail. MDM.
Mise en oeuvre pratique des différentes phases :
nettoyage, enrichissement, organisation des données.

Atelier : construction d'un système de détection de contours

Apprendre à mettre en place un modèle d'apprentissage simple

Les différentes méthodes : apprentissage supervisé, apprentissage automatique.
Algorithmes : régression linéaire, k-voisins, classification naïve bayésienne, arbres de décision, ...

Atelier : classifieurs. scoring

APIs fournies en standard, modèles d'apprentissage
Projet scikit-learn : classification, régression, validation de modèles prédictifs.
Démonstrations avec les modèles fournis par scikit-learn
Positionnement et comparaison avec Keras, mxnet, caffe

Atelier : codage d'une reconnaissance d'animaux avec une forêt aléatoire

Choisir entre la régression et la classification en fonction du type de données

Présentation des types de données en entrées : données discrètes, données continues. Labelisation, mapping par fonction.
Comprendre les algorithmes : régression linéaire, k-moyennes, k-voisins, classification naïve bayésienne, arbres de décision, forêts aléatoires, ...
Critères de choix des algorithmes.

Atelier : construction d'un système décisionnel fondé sur des forêts aléatoires

Évaluer les performances prédictives d'un algorithme

Les courbes d'apprentissage. Définitions : AUC, courbes ROC.
Comprendre le principe des hyper-paramètres. Choix des hyper-paramètres.

Atelier : calcul et visualisation d'une matrice de confusion

Atelier : Visualisation de courbes d'apprentissage fonction des hyper-paramètres

Atelier : Visualisation d'une mise en sur-apprentissage



Phirio

Créer des sélections et des classements dans de grands volumes de données pour dégager des tendances

Présentation de pyspark
Machine learning et deep learning
TensorFlow: principe de fonctionnement, plateformes supportées, distribution,

Cassandra

Cassandra

Cassandra

Introduction
Historique, fonctionnalités de Cassandra, licence
Format des données, "key-value", traitement de volumes importants,
haute disponibilité, système réparti de base de données, ...

Installer et configurer le SGBD NoSQL Apache Cassandra

Installation et configuration
Prérequis. Plateformes supportées. Etude du fichier de configuration : conf/cassandra.yaml
Répertoire de travail, de stockage des données, gestion de la mémoire.

Atelier : démarrage d'un noeud et test de l'interface cliente cqlsh.

Appréhender le CQL (Cassandra Query Language)

Commandes de base : connexion au système de base de données,
création de colonnes, insertion, modification recherche,
Le CQL : Cassandra Query Language.
Limitations du CQL.

Créer une base de données et manipuler ses objets

Utilisation de Cassandra
Création de bases et interrogation avec cql
Définition de la notion de consistance. Eléments en jeu : Commit.log, Memtable, Quorum
Comment écrire des requêtes ? Approches.

Atelier : premiers pas avec une base de données Cassandra pré-chargée
mise à disposition sur l'infrastructure de travaux pratiques



Phirio

Connaitre la notion de grappe au sein de la base de données

Gestion de la grappe.
Principe. Configuration des noeuds.
Notion de bootstrapping et de token.
Paramètres de démarrage des noeuds.
Réplication: topologie du réseau et EndpointSnitch.
Stratégie de réplication.
Méthode d'ajout de noeuds et suppression.
Architecture de stockage mémoire et disque dur, gestion des tombstones, bloom-filter

Atelier : mise en place d'une configuration de production (multi-datacenters, multi-racks)

Administrer et sécuriser un cluster Cassandra

Exploitation.
Gestion des noeuds Cassandra.
Sauvegardes, snapshots et export au format JSON.
Principe de cohérence, hinted_handoff, digest request et read repair.
Sécurité

Atelier : paramétrage, authentification et sécurisation de la base system_auth.

Gestion des rôles et des autorisations sur une application standard.

Support Hadoop et Spark

Principe de map/reduce. Implémentation Hadoop et intégration Hadoop/Cassandra.
Support Spark :

Description rapide de l'architecture spark.

Atelier : Mise en oeuvre depuis Cassandra. Execution d'application Spark s'appuyant sur une grappe Cassandra.

Supervision et performances

Prometheus: apports et particularité de prometheus pour la supervision cassandra
Supervision avec nodetool.
Principe des accès JMX , exports JMX vers des outils de supervision.

Atelier : démonstration avec Prometheus et Grafana.

Performance :
Présentation de l'outil de test de performance Cassandra-stress

Atelier : mise en place d'un plan de stress et paramétrage.

Spark

Spark



Phirio

Spark

Présentation Spark, origine du projet, apports, principe de fonctionnement. Langages supportés.
Modes de fonctionnement : batch/Streaming.
Bibliothèques : Machine Learning, IA
Mise en oeuvre sur une architecture distribuée. Architecture : clusterManager, driver, worker, ...
Architecture : SparkContext, SparkSession, Cluster Manager, Executor sur chaque noeud. Définitions : Driver program, Cluster manager, deploy mode, Executor, Task, Job

Savoir intégrer Spark dans un environnement Hadoop

Intégration de Spark avec HDFS, HBase,
Création et exploitation d'un cluster Spark/YARN. Intégration de données sqoop, kafka, flume vers une architecture Hadoop et traitements par Spark.
Intégration de données AWS S3.
Différents cluster managers : Spark interne, avec Mesos, avec Yarn, avec Amazon EC2

Atelier : Mise en oeuvre avec Spark sur Hadoop HDFS et Yarn. Soumission de jobs, supervision depuis l'interface web

Développer des applications d'analyse en temps réel avec Spark Structured Streaming

Objectifs , principe de fonctionnement: stream processing. Source de données : HDFS, Flume, Kafka, ...
Notion de StreamingContext, DStreams, démonstrations.

Atelier : traitement de flux DStreams en Scala. Watermarking. Gestion des micro-batches.

Intégration de Spark Structured Streaming avec Kafka

Atelier : mise en oeuvre d'une chaîne de gestion de données en flux tendu : IoT, Kafka, Spark Structured Streaming, Spark. Analyse des données au fil de l'eau.

Faire de la programmation parallèle avec Spark sur un cluster

Utilisation du shell Spark avec Scala ou Python. Modes de fonctionnement. Interprété, compilé.
Utilisation des outils de construction. Gestion des versions de bibliothèques.

Atelier : Mise en pratique en Java, Scala et Python. Notion de contexte Spark. Extension aux sessions Spark.



Phirio

Manipuler des données avec Spark SQL

Spark et SQL

Traitement de données structurées. L'API Dataset et DataFrames
Jointures. Filtrage de données, enrichissement. Calculs distribués de base. Introduction aux traitements de données avec map/reduce.

Lecture/écriture de données : Texte, JSON, Parquet, HDFS, fichiers séquentiels.

Optimisation des requêtes. Mise en oeuvre des DataFrames et DataSet. Compatibilité Hive

Atelier : écriture d'un ETL entre HDFS et HBase

Atelier : extraction, modification de données dans une base distribuée.
Collections de données distribuées. Exemples.

Support Cassandra

Description rapide de l'architecture Cassandra. Mise en oeuvre depuis Spark. Exécution de travaux Spark s'appuyant sur une grappe Cassandra.

Spark GraphX

Fourniture d'algorithmes, d'opérateurs simples pour des calculs statistiques sur les graphes

Atelier : exemples d'opérations sur les graphes.

Avoir une première approche du Machine Learning

Machine Learning avec Spark, algorithmes standards supervisés et non-supervisés (RandomForest, LogisticRegression, KMeans, ...)

Gestion de la persistance, statistiques.

Mise en oeuvre avec les DataFrames.

Atelier : mise en oeuvre d'une régression logistique sur Spark

Neo4j

Neo4j

Neo4j

Présentation Neo4j, les différentes éditions, license

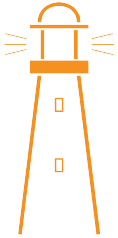
Fonctionnalités, stockage des données sous forme de graphes

CQL : Cypher Query Language

Positionnement par rapport aux autres bases de données, apports de Neo4j

L'analyse de données.

Cas d'usage



Phirio

Installation et configuration

Les différentes méthodes d'installation

Atelier : installation de Neo4J Enterprise Edition en cluster.

Premiers pas avec l'interface web.
Création de données, requêtage
Import de données

Cypher Query Language

Syntaxe, description des relations avec CQL, les patterns
Les clauses d'écriture : set, delete, remove, foreach,
de lecture : match, optional match, where, count, case, ...
Les fonctions : count, type, relationship, ...
Principe de profondeur et de direction de relation dans une recherche
Les listes et les projections maps
Les algorithmes de Graphe

Atelier : création d'un graphe,

Requêtes de recherche, navigation dans le graphe

Exploitation

Sauvegardes et restaurations
Optimisation des transactions
Indexation
Client jmx
Points de surveillance

Développement

Description des APIs disponibles: .Net, Java, Javascript, Python
Connexions, sessions et transactions
Principe de causalité entre transactions
La bibliothèque Apoc

Atelier : connexion et récupération de données provenant de Cassandra

Sécurité

Principe et activation
Paramétrage

Atelier : création d'un compte sécurisé

Kafka

Kafka



Phirio

Kafka

Le projet Kafka : historique, fonctionnalités, principe de fonctionnement.
Présentation de l'architecture et du rôle de chaque composant : broker, producer, consumer
Liaison éventuelle avec Zookeeper. Impacts.

Acquérir les bonnes pratiques de distribution de messages

Etude de la configuration du broker

Atelier : création d'une configuration multi-broker, démarrage de plusieurs noeuds

Atelier : création d'un topic simple et mise en oeuvre d'une chaîne de base. Visualisation des messages avec kafka-console-consumer

Savoir configurer Kafka pour intégrer les données de différents formats et de sources différentes

Kafka Connect : présentation des fonctionnalités : intégration de données d'origines multiples, modes de fonctionnement (standalone ou distribué)
Types de connecteurs

Atelier : configuration de connecteurs, ingestion de données, création d'une chaîne de transformation

Appréhender les différentes APIs de Kafka.

Conception d'applications avec Kafka. Principe de fonctionnement.

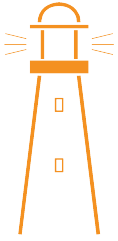
Atelier : développement de prototypes en Python, Java, Scala

Couplage avec SparkStreaming en mode batch, en mode continu
Principe et architecture de Kafka Streams

Mettre en oeuvre KSQL

Conception d'application avec KSQL. La sur-couche KSQL.
Présentation de ksqlDB. Création de flux en KSQL. Les ajouts SQL pour permettre le traitement en flux.
Micro-batches. Introduction au water-marking.

Atelier : mise en oeuvre d'une chaîne de traitement avec KSQL



Phirio

Travailler en sécurité avec Kafka

Intégrité et confidentialité :
Chiffrement SSL et authentification SASL.
Sécurisation de l'infrastructure : Zookeeper, proxy Rest
Disponibilité : La réplication. Facteur de réplication. Partitions

Atelier : tests de haute disponibilité dans une configuration multi-noeuds

Exploitation

Mise en oeuvre de kafka-tool
Gestion des logs
Détection de pannes : méthodes et outils
Benchmarks

Intégration SQL

Intégration SQL

Intégration SQL

Besoin. Adéquation entre les objectifs et les outils.
Faciliter la manipulation de gros volumes de données en conservant une approche utilisateurs.
Rappels sur le stockage : HDFS, Cassandra, HBase
et les formats de données : parquet, orc, raw, clés/valeurs
Les outils : Hive, Impala, Tez, Presto, Drill, Phoenix, Spark-sql, Spark Dataframe

Hive

Présentation Hive. Mode de fonctionnement. Rappel sur map/reduce.
Hive : le langage HiveQL. La surcouche Tez.

Atelier : création de tables, requêtage, connexion avec Hbase.

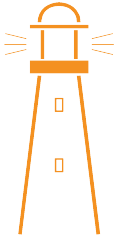
Impala et Phoenix

Présentation Impala. Cadre d'utilisation. Contraintes. Liaison avec le métastore Hive.

Atelier : mise en évidence des performances.

Présentation Phoenix. Cadre d'utilisation. Contraintes.

Atelier : connexion et requêtage sur une table Hbase.



Phirio

Presto

Cadre d'utilisation. Sources de données utilisables.

Atelier : mise en oeuvre d'une requête s'appuyant sur Cassandra et PostgreSQL.

Spark-sql et Spark DataFrame

Les différentes approches. Syntaxe Spark-sql, Spark/SQL. APIs QL. Utilisation du métastore Hive.

Atelier : mise en oeuvre d'une requête s'appuyant sur une table HBase et sur HDFS. Requêtage en spark-sql sur un fichier csv.

Drill

Utilisation d'APIs JDBC, ODBC. Indépendance Hadoop. Contraintes d'utilisation. Performances.

Atelier : lecture de fichiers Parquets dans du HDFS, jointures, connexion et requêtage sur une table Hbase.

Comparatifs

Compatibilité ANSI/SQL. Approches des différents produits. Critères de choix.

Serious game : développement d'un prototype

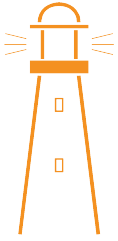
La société DataIA rencontre de sérieuses difficultés avec sa toute nouvelle infrastructure BigData : le traitement des données est très lent, ils n'arrivent à aucun résultat dans un temps correct. Ils décident de se faire aider par une équipe d'administrateurs pour analyser et surveiller les différents composants de leur architecture afin d'en trouver les failles et de proposer des correctifs. Cette équipe saura-t-elle relever le défi ?

La méthode

Simulation d'un cas d'étude, avec un travail collaboratif sur des données réelles, accessibles en opendata, et des labs techniques (Prometheus, Graphite, Influxdb, ElasticSearch, ..)
Épreuves personnelles et épreuves en commun vont permettre de contrôler les connaissances et d'échanger entre stagiaires, tout en bénéficiant du soutien et des explications complémentaires du formateur sur les thèmes proposés

Les jeux

Battle d'architecture, la techno mystère, l'intrus, les points de faiblesse, etc...



— Phirio —

Le debrief

Retour des travaux, bilan des points individuels et classement des joueurs.
Retour d'expérience des participants