



## Dask : mise en oeuvre, programmation

CB042

Durée: 3 jours

2 510 €

27 au 29 janvier  
14 au 16 avril

7 au 9 juillet  
3 au 5 novembre

### Public :

Chefs de projet, Data Scientists, Développeurs, Architectes...

### Objectifs :

Savoir mettre en oeuvre Dask pour paralléliser des calculs en Python

### Connaissances préalables nécessaires :

Bases de la programmation python.

### Objectifs pédagogiques :

### Programme :

#### Introduction

Présentation de Dask, fonctionnalités, apports. Comparaison avec d'autres environnements : yarn, spark.  
Calculs parallèles en environnements distribués, ou sur un seul serveur.  
Les composants de Dask : scheduler, collections BigData.

#### Premiers pas avec Dask

Différentes méthodes d'installation : Anaconda, pip, depuis les sources

Atelier : installation, et création d'objets Dask,

choix des méthodes et tâches, visualisation des graphes d'exécution.  
exécution par le scheduler



# Phirio

---

## Elements de base

---

Array: cas d'usages, compatibilité NumPy, définition de chunks, exemples, bonnes pratiques

Atelier : création, stockage de Dask Array

Bag : définition, limites

Atelier : exemple de création, stockage, calcul sur des Dask Bags

Dask Dataframes : regroupement de dataframes pandas, stockage sur disque ou dans un cluster, critères de choix par rapport aux dataframes pandas, bonnes pratiques, compatibilité avec Parquet, intégration de tables SQL

Atelier : mise en oeuvre de dask.dataframes et comparaison avec pandas

Delayed ou Futures : une exécution stockée dans un graphe d'actions, ou en temps réel, critères de choix

---

## Fonctionnement avancé

---

Gestion des performances  
Configuration du scheduler  
Les graphes d'exécution  
Utilisation du dashboard  
Outils de debugging

Atelier : tests de performances et debugging

---

## Dask.distributed

---

Fonctionnalités : exécution dans un environnement distribué ou en local, outils de diagnostic et de suivi des performances, utilisation de l'API Futures pour des calculs en temps réel  
Architecture : dask-scheduler et dask-worker

Atelier : mise en oeuvre de dask.distributed : installation, configuration, initialisation d'un client.

Présentation du dashboard  
Analyse des performances  
Limites de Dask.distributed  
Bonnes pratiques

---

## Dask-ML

---

Apports : utiliser les outils classiques de machine learning comme scikit-learn dans un environnement Dask  
Exemples d'utilisation : modèles complexes, volumes de données importants  
Présentation de Dask-ML et principe de fonctionnement  
Intégration scikit-learn, PyTorch, Keras / Tensorflow

Atelier : Installation et exemples avec scikit-learn