TUTORIEL D'UNE TECHNOLOGIE ÉMERGENTE : SPARK

SOMMAIRE

INTRODUCTION

- Framework open-source de calcul distribué
- 1ère version en 2014
- Projet très populaire de la fondation Apache
- Cadre applicatif de traitements Big Data
- Optimisation de l'utilisation des ressources d'un cluster de machines

CONTEXTE

- Nécessité d'un moteur de calcul et d'un modèle de programmation distribué pour des raisons économiques
- Historiquement, les processeurs étaient de plus en plus rapide donc les applications étaient principalement exécutées sur un seul processeur
- 2005 : Limitations au niveau de la dissipation de chaleur ne permettant plus d'augmenter la cadence des processeurs
- Choix d'augmenter le nombre de cœurs au processeur qui a pour effet de revoir les patrons utilisés pour la création d'applications

CONTEXTE

- Dans un même temps, le coût des technologies de stockage et d'acquisition de données ont considérablement diminué
- Explosion de la quantité de données disponible
- Création d'un nouveau besoin d'analyse de ces mégadonnées dites « Big Data »

HISTORIQUE

- 2009 : Conception de Spark par Matei Zaharia lors de son doctorat
- 2013 : Transmission de Spark à la fondation Apache
- 2014 : Spark remporte le Daytona GraySort Contest (trier 100 To de donnés le plus rapidement possible) avec un record de seulement 23 minutes

PHILOSOPHIE : PLATEFORME UNIFIE

- Création d'application Big Data
- Grande variété de tâches : importation de données et requêtes, apprentissage automatique et calculs sur des flux (streaming)
- Même moteur de calcul pour ces différentes tâches
- Supporte différents langages via ses API de haut niveau : Scala, Java, Python, SQL et R

PHILOSOPHIE : MOTEUR DE CALCUL

- Ensemble d'opérations se font en mémoire RAM
- Ne prend pas en charge le stockage des données après les calculs
- Mais il reste compatible avec différentes solutions telles que Amazon S3, Apache Cassandra, Apache Hadoop ou Apache Kafka
- Se distingue ainsi de Hadoop avec son système de fichiers distribués (HDFS)

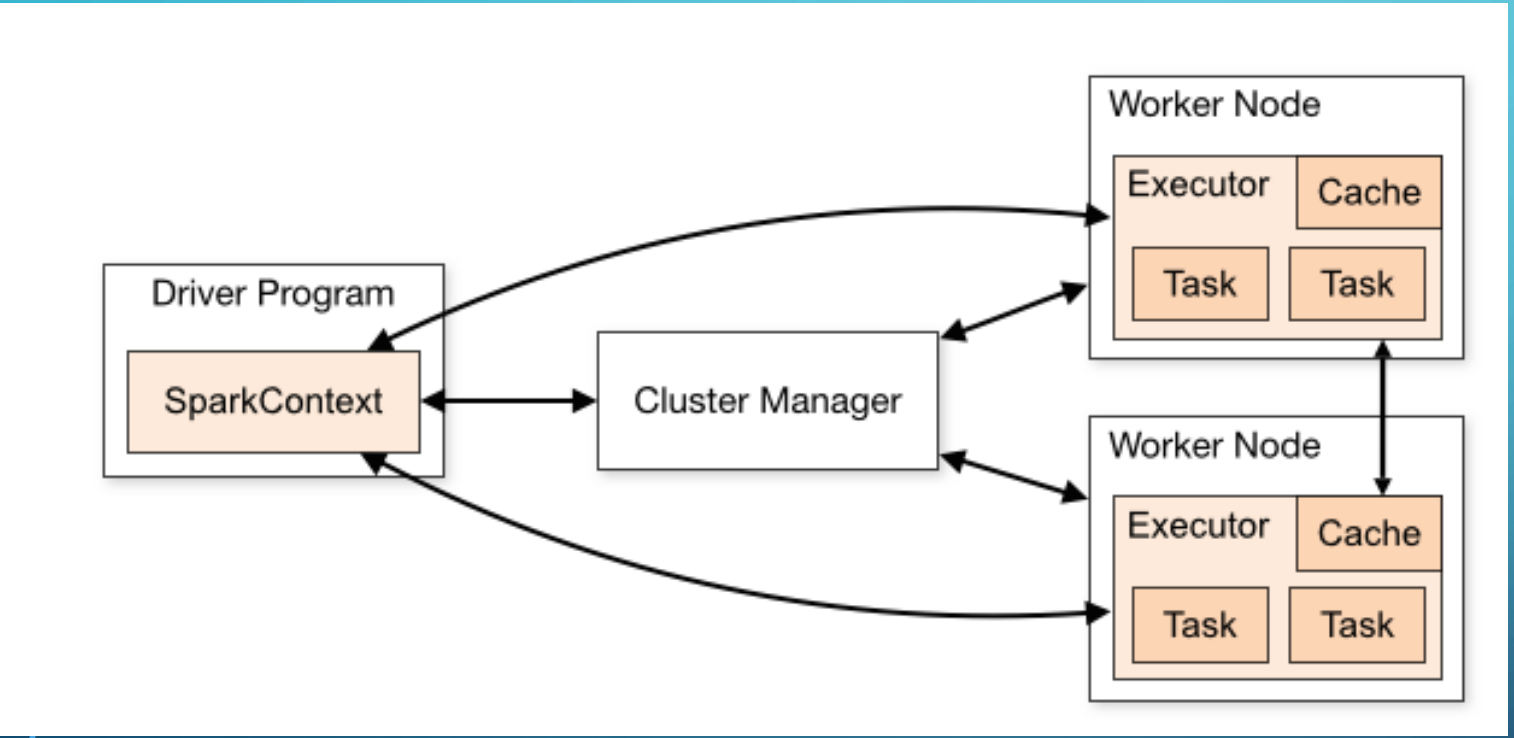
PHILOSOPHIE : LIBRAIRIES

- Un des aspects les plus importants de Spark
- Différentes fonctionnalités :
  - Requête sur des données structurées (Spark SQL)
  - Apprentissage automatique (Spark MLib)
  - Traitement en continu de flux de données (Spark Streaming)
  - Analyse de graphe (Spark Graph X)

ARCHITECTURE

- Spark contrôle et coordonne l'exécution de tâches sur un cluster de machines
- Un cluster est composé de :
  - nœud driver : chargé d'analyser, distribuer et programmer les tâches sur les différents exécuteurs
  - un ou plusieurs nœuds workers avec chacun un exécuteur qui est chargé d'exécuter les tâches confiées par le driver
  - un cluster manager qui est chargé d'instancier les workers

ARCHITECTURE



Plusieurs types de cluster manager :

- Mode autonome de Spark ("standalone")
- Apache Mesos
- Apache Hadoop YARN

Supporte scalabilité et échec de tâches