OpenStack/Hadoop/Spark LSST

LALDev
18 mai 2016

Christian Arnault
Axel Chevarin

Contexte

- BigData pour l'astro
- Le pipeline de production des données SDRP pour LSST
  - Split Data Release Production (NCSA – CCIN2P3)
  - L'application « Stack »
- Le modèle du Stack n'est pas scalable
  - architecture linéaire
- => *technologie Spark ???*
- Cf la journée Loops
- Intérêt dans l'environnement VirtualData/OpenStack au LAL
  - Mise en service de la plateforme OpenStack
  - Apprentissage !!

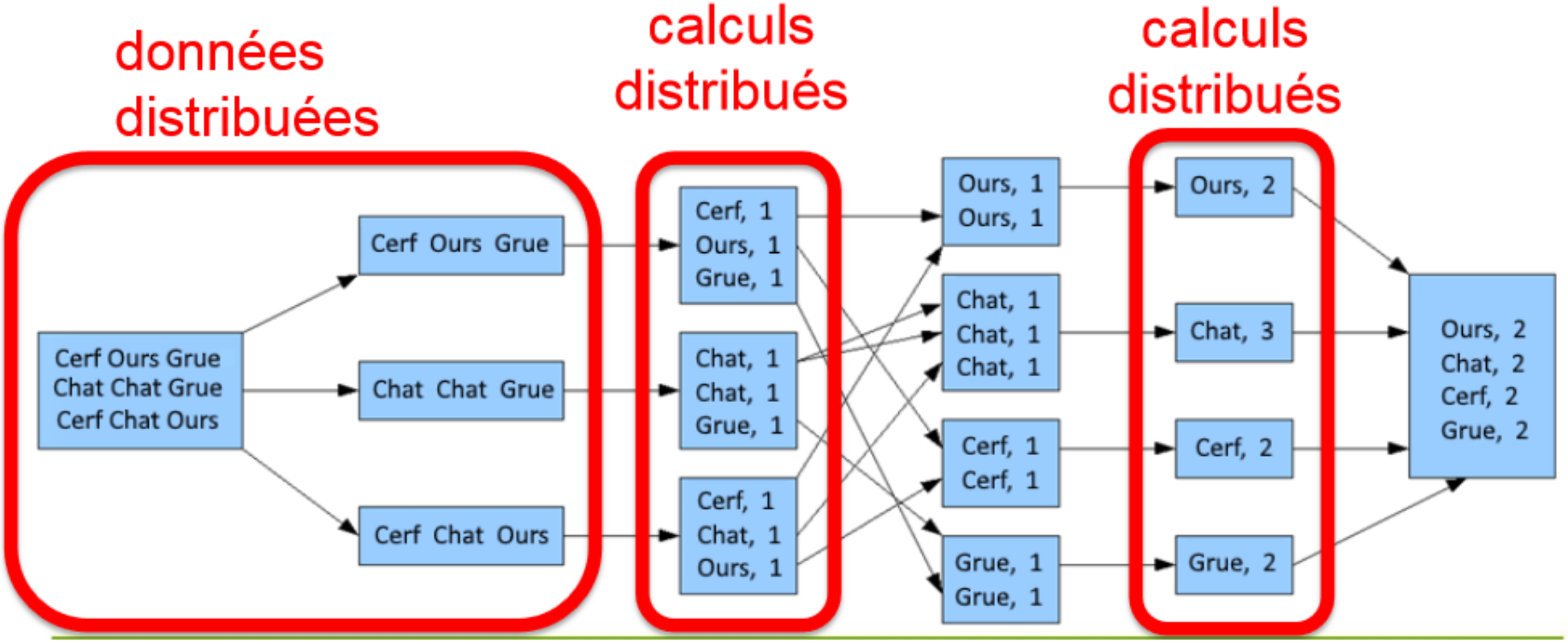
LSST

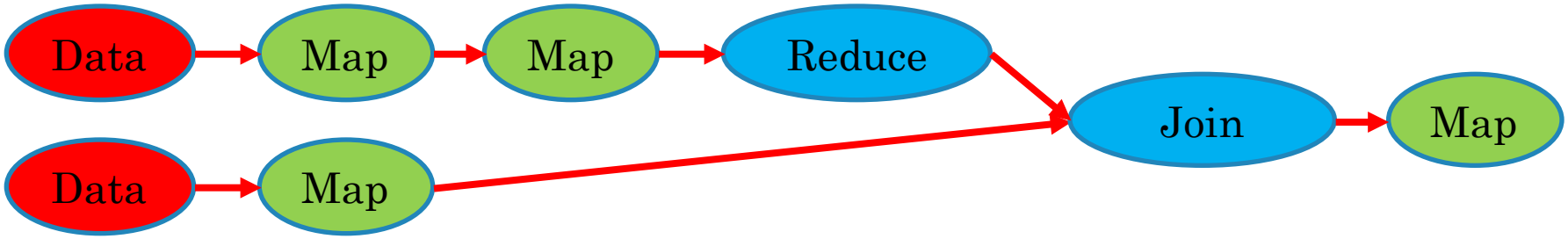
- Plusieurs sujets où la techno pourrait s'appliquer:
  - Gestion des très grand nombre de fichiers FITS (> 10^9)
  - Indexation des fichiers FITS
    - Indexation géographique 2D (coordonnées galactique)
    - Indexations logiques (CCD, Flags de production…)
  - Logique de pipeline (enchaînement de processus)
  - Machine learning pour la catégorisation des objets du ciel
  - Très grandes images + zooming
- Liaison avec le projet Petasky
  - Études sur les big data en astro
    - noSQL, architectures de distribution
- Mesures de performances

Hadoop/Spark

- (*cf la journée Loops pour les détails*)
- Le principe de l'algorithmique fonctionnelle Map/Reduce
  - Les données:
    - ensemble de couples (Clés/Valeurs)
  - Les actions principales:
    - MAP: transformation d'une donnée
    - REDUCE: assemblage (réduction) d'un ensemble de données
  - Les actions secondaires:
    - Filtre: sélection d'un sous-ensemble
    - Opérations ensemblistes: Join, Exclusion, Intersection, …
  - Les applications:
    - Une application = Direct Acyclic Graph d'actions élémentaires

Map/Reduce


données distribuées
calculs distribués
calculs distribués
Cerf Ours Grue
Chat Chat Grue
Cerf Chat Ours
Cerf Ours Grue
Chat Chat Grue
Cerf Chat Ours
Cerf, 1
Ours, 1
Grue, 1
Chat, 1
Chat, 1
Grue, 1
Cerf, 1
Chat, 1
Ours, 1
Ours, 1
Ours, 1
Chat, 1
Chat, 1
Chat, 1
Cerf, 1
Cerf, 1
Grue, 1
Grue, 1
Ours, 2
Chat, 3
Cerf, 2
Grue, 2
Ours, 2
Chat, 2
Cerf, 2
Grue, 2

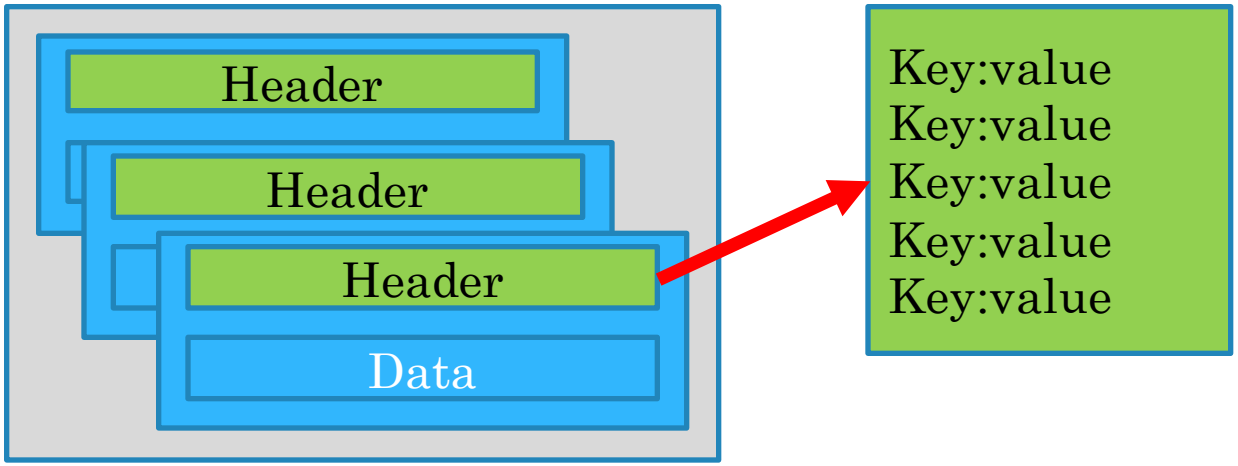

Data
Map
Map
Reduce
Data
Map
Join
Map

Hadoop/Spark

- Hadoop:
  - FileSystem distribué (HDFS)
  - + Plusieurs modules fonctionnels
    - MapReduce
    - Formattage des données
    - Gestion d'applications
- Spark
  - Hadoop + algorithmes d'optimisation du DAG M.R.
    - Distribution des données en terme de RDD
      - Resilient Distributed Datasets
    - La connaissance du DAG permet d'optimiser
      - La distribution des données
      - La distribution des processus
      - L'utilisation de la mémoire vs. le disque

Mise en place de l'infrastructure

- Cluster de VMs sur OpenStack
- Hadoop/Spark fournis par Apache
- Installation
  - sur une machine Ubuntu
  - Toute la suite Numpy + Matplotlib + Scipy
- Données astro au CCIN2P3 :
  - CFHT: Hawaï
- Le format de base des données: fichier FITS



Sujets

- Indexation géographique des données FITS
  - Chaque fichier fournit les données (images) pour
    - UN CCD
    - UNE prise de données
  - Chaque Header du fichier FITS contient les métadonnées de la prise de vue (orientation du téléscope, date, conditions, …)
  - Problème:
    - Trouver les fichiers FITS correspondants à une zone du ciel
      - Coordonnées galactique (Ascension, Déclinaison)
    - Et associés à un ensemble de conditions donné (date, qualité, …)
    - Dans un « run »
      - » 10^9 fichiers FITS

Sujets …

- Simulation: Production de catalogues d'objets astrophysiques
  - Étoiles, galaxies
  - Un modèle physique => un catalogue de galaxies (>> 10^9 galaxies)
  - Étude et comparaison avec les observations

- Mesures de performance:
  - Suivi des facteurs d'échelles
    - Nombre de workers pour distribuer les données et les processus
  - Configuration des machines
    - Mémoire
    - Cores

Sujets …

- Apprentissage de la technologie
  - Langage de programmation
    - Scala
    - Python
  - Modèle fonctionnel

- La structure et le format des données
  - Spark travaille sur la base de données structurées (ensembles de clés – valeurs)
  - Très bien adapté à des formats textes et tabulaires (ex CSV, SQL, …)
  - Mais les formats binaires doivent être expliqués à Spark
    - Description par des schémas
    - Librairies spécialisées
      - Hadoop/avro

Status

- La plateforme de base est installée
  - Ubuntu
  - 1 VM pour HDFS
  - 8 VM pour Spark
    - 1 master
    - 7 workers
  - Les VMs: 4 cores, Ram:8Go, Disque:60Go
  - Installation de la suite Numpy, Scipy, Matplotlib