KARMA

Le système de Revenue Management d'Air France KLM avec Hadoop

Conférence BIG DATA - Master MBDS
Université de Nice Sophia Antipolis

16 Décembre 2014

Martial AYAS – maayas@airfrance.fr

KARMA

AIRFRANCE KLM

Agenda

KARMA

1. Présentation de KARMA
   - Définitions et concepts
   - Quelques chiffres
2. L'utilisation d'Hadoop dans KARMA
   - Le choix d'Hadoop
   - Design technique d'un batch Hadoop
3. Cas d'utilisation
   - Mise à jour du programme de vol
   - Optimisation des traitements
   - Equilibrage des traitements
4. Axes d'amélioration et évolutions
   - Performances et accès aux données
   - Migration vers Hadoop 2

Présentation de KARMA

Définitions et concepts

KARMA

- KARMA ➔ KLM Air France Revenue Management Application
- RMS : Revenue Management System
- KARMA permet d'optimiser le revenu grâce à la prévision :
  - Demande
  - Annulation
  - Overbooking
- Permet aux analystes de vols d'agir sur les recommandations du système en fonctions :
  - Des marchés
  - Des périodes
  - Des évènements

L'objectif principal de KARMA

« To sell the right seat to the right person at the right moment »

Et d'influer sur la disponibilité des sièges à vendre pour un tarif donné à une date donnée.

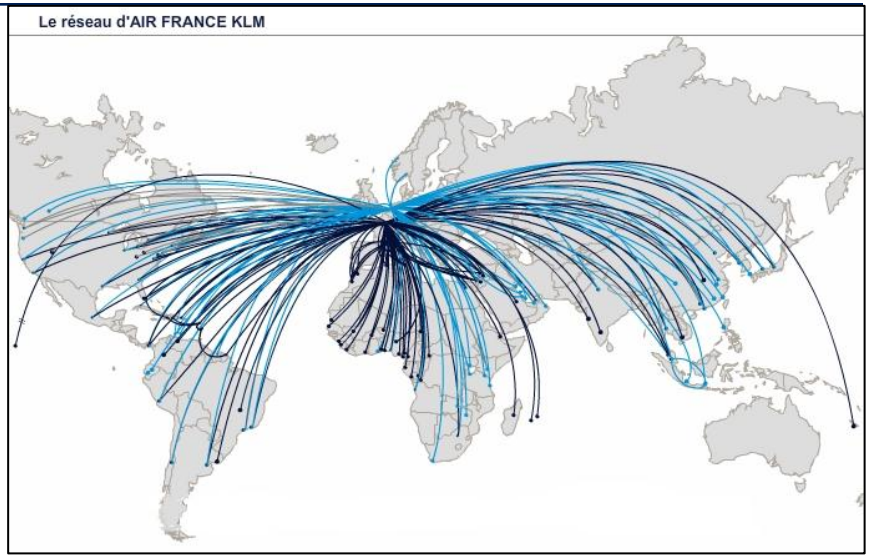
AIRFRANCE KLM

Présentation de KARMA

Chiffres clés

KARMA

- Le programme de vol
  - 2500 vols / jour
  - 231 destinations / 103 pays
- L'activité « Passage » (2013)
  - 78 millions de passagers
  - Profils Loisir / Business / Groupes
- Les avions
  - 552 avions
  - De 40 à 520 sièges
- La tarification
  - 26 classes
  - 30 000 tarifs
  - Prix carburant
  - La concurrence
  - La demande



➢ Volumes initiaux + Combinatoire = **Démultiplication des volumes**

➢ Augmentation des volumes = **Problématiques de performances**

Utilisation d'Hadoop dans KARMA

Le choix d'Hadoop (1/2)

KARMA

- Le choix Hadoop part d'un constat :
  - Refonte du RM = Nouvelle approche ➔ Nouveaux besoins
    - Forte augmentation des volumes nécessaires aux calculs de prévision
    - Forte augmentation de la fréquence des évènements à prendre en compte pour avoir un système réactif

    ➔ Augmentation des volumes et du stockage

    ➔ Augmentation de la puissance de traitement
  - Infaisabilité des traitements batch en BD
    - Le volume des données lié aux évènements et aux combinaisons possibles
    - Temps de traitements incompatibles avec la fréquence nécessaire des MAJ

    ➔ Nécessité de paralléliser et distribuer les traitements et les données
  - Nécessité de communiquer avec le composants de RO
    - Les moteurs de Recherche Opérationnelle (Prévision + Optimisation)
    - Développés en C++ (CPlex), utilisent l'approche Map Reduce / Fichiers

    ➔ L'alimentation des moteurs doit se faire sous la forme de fichiers

Utilisation d'Hadoop dans KARMA

Le choix d'Hadoop (2/2)

KARMA

- Le choix Hadoop n'est pas sans contraintes …
  - Contraintes d'exploitation
    - Haute disponibilité / Tolérance aux pannes
    - Seuls la Base de données et le NFS sont supportés (backup) par la production
      ➔ Mécanisme de synchronisation entre la BD, le NFS et le HDFS
    - Maitrise du stockage malgré les volumes ➔ Utilisation du format Avro + compression
  - Contraintes de développement / Maintenabilité
    - Apprentissage de l'approche et des API
    - Démultiplication des composants : 1 req SQL ➔ 1 à n jobs Hadoop

    Ex : Le traitement d'optimisation quotidien comporte 600 jobs dont 350 en Hadoop

    ➔ Création d'un Framework de développement pour harmoniser le dev des jobs Hadoop
    - Chaque job de MapReduce crée une nouvelle copie des données

    ➔ Besoin d'un outil de design des jobs est de suivi du cycle de vie des traitements et des données produits par Hadoop
  - Contraintes métier
    - Concilier les 3 activités : Batch / Utilisateur / Evènementielle (Problématique de concurrence sur l'accès et la mise à jour des données)

    ➔ Séparation des traitements par la planification / contraintes temporelles importantes

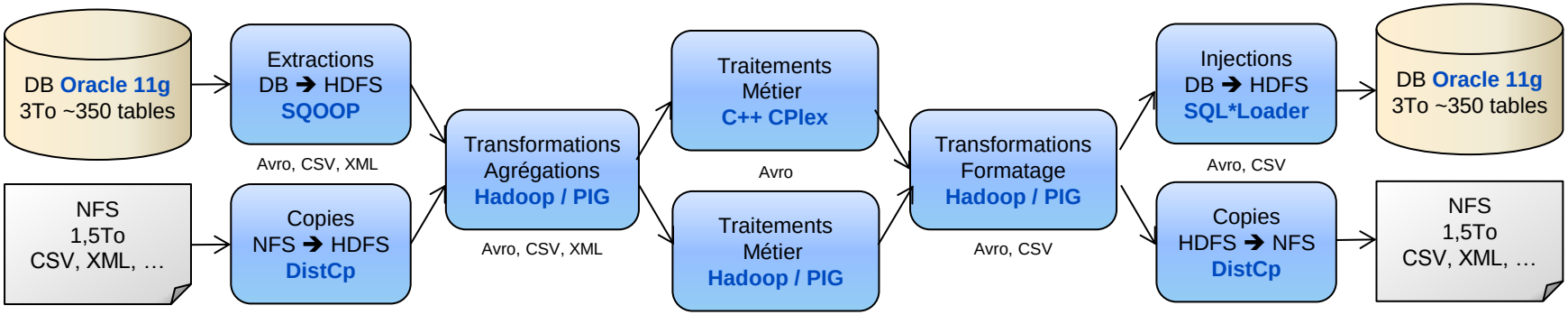
    ➔ Dénormalisation du modèle de données

AIRFRANCE KLM

Utilisation d'Hadoop dans KARMA

Design technique d'un Batch Hadoop

KARMA



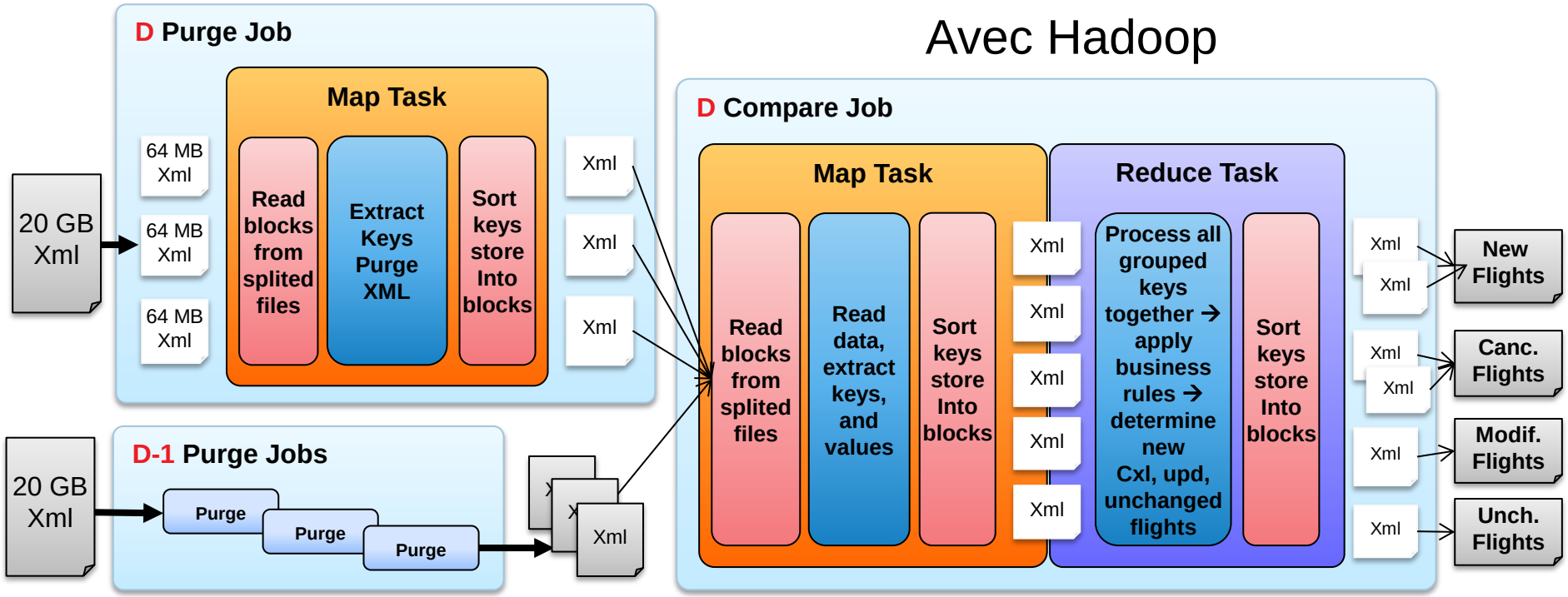
- Accès aux données DB
  - SQOOP + OraOop : Extractions en //
  - SQL*Loader : Injection en //

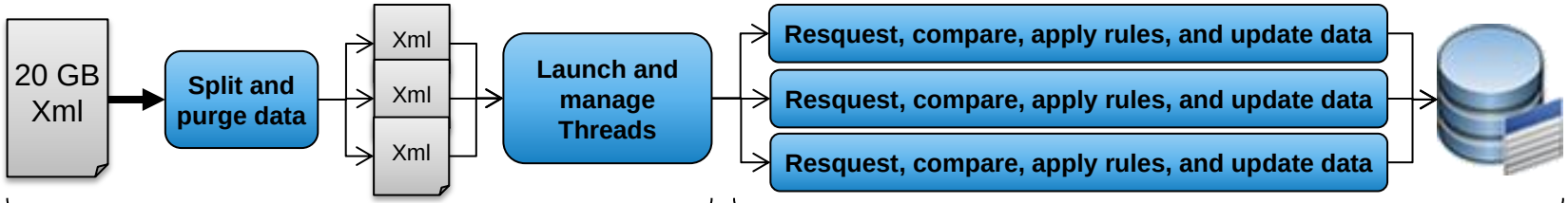
  (Suppression des contraintes d'intégrité, dénormalisation, recalcul des indexs)
- Accès aux données NFS
  - DistCp : Copies HDFS ←→ NFS
- Préparation des données
  - Hadoop MapReduce : Transformation, comparaisons, jointures, agrégations, filtres, fusion, …
- Traitements métier
  - Hadoop MapReduce : traitements hors RO
  - C++ Cplex : Moteurs de prévisions, d'optimisations (Recherche Opérationnelle)
- Reporting
  - PIG : Statistiques et KPI vérification de la qualité des données
- Exploitation / Supervision
  - Error collector, compacteurs, archivage, purge

Cas d'utilisation

Mise à jour du programme de vol

KARMA





Avec un SGBD

Complexité à la charge du développeur

La quasi-totalité du traitement repose sur la BD → PB Contention → PB Puissance

Actions implemented by Developers **Actions supported by Hadoop Framework**

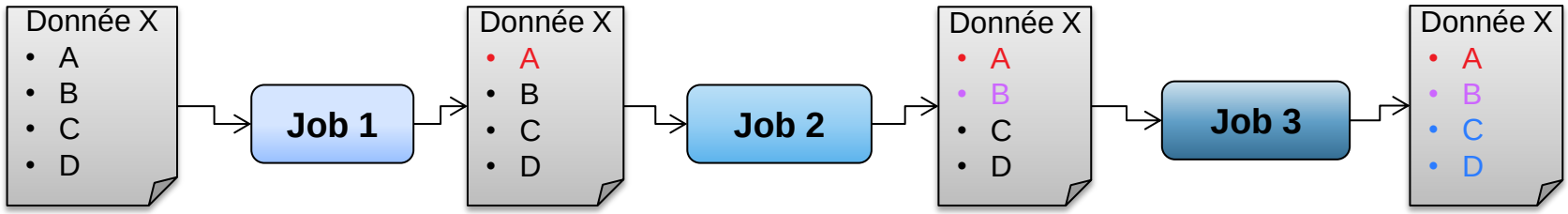
Cas d'utilisation

Optimisation des traitements (1/2)

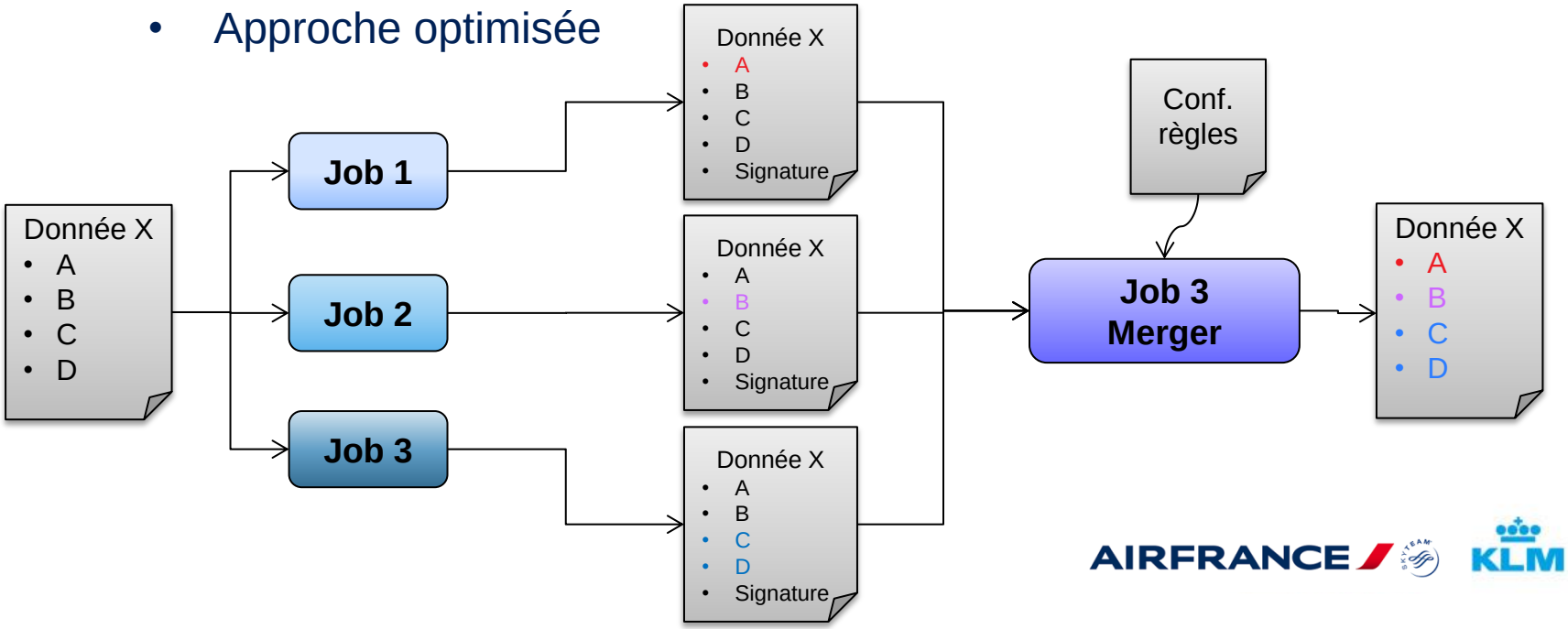
KARMA

Décomposer pour mieux paralléliser

- Approche séquentielle



- Approche optimisée



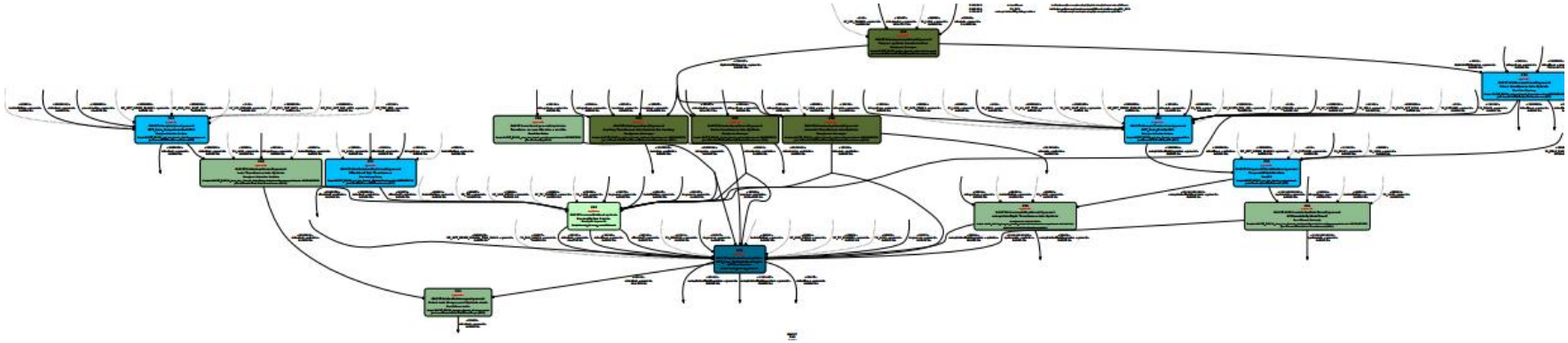
AIRFRANCE KLM

Cas d'utilisation

Optimisation des traitements (2/2)

KARMA

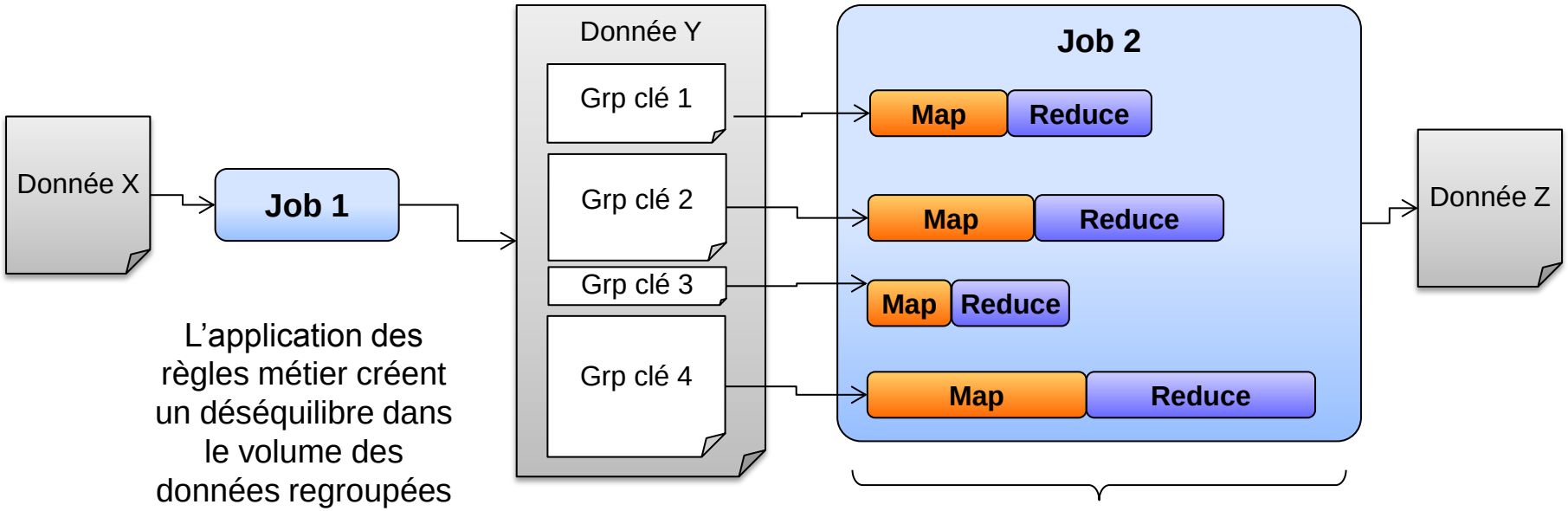
- Optimiser par le séquencement et la disponibilité des données
  - Démultiplication du nombre de jobs Hadoop
  - Analyse des entrées / sorties de chaque jobs
  - Déterminer le séquencement optimal des traitements
  - Les traitements se lancent uniquement lorsque l'ensemble des données sont prêtes
  - Plus il y a de traitements parallélisés mieux on utilise la grille



Cas d'utilisation

Equilibrage des traitements (1/2)

KARMA



L'application des règles métier créent un déséquilibre dans le volume des données regroupées selon la clé choisie

Le temps de traitement est égal au temps du job le plus long (dont le volume de données à traiter et le plus important)

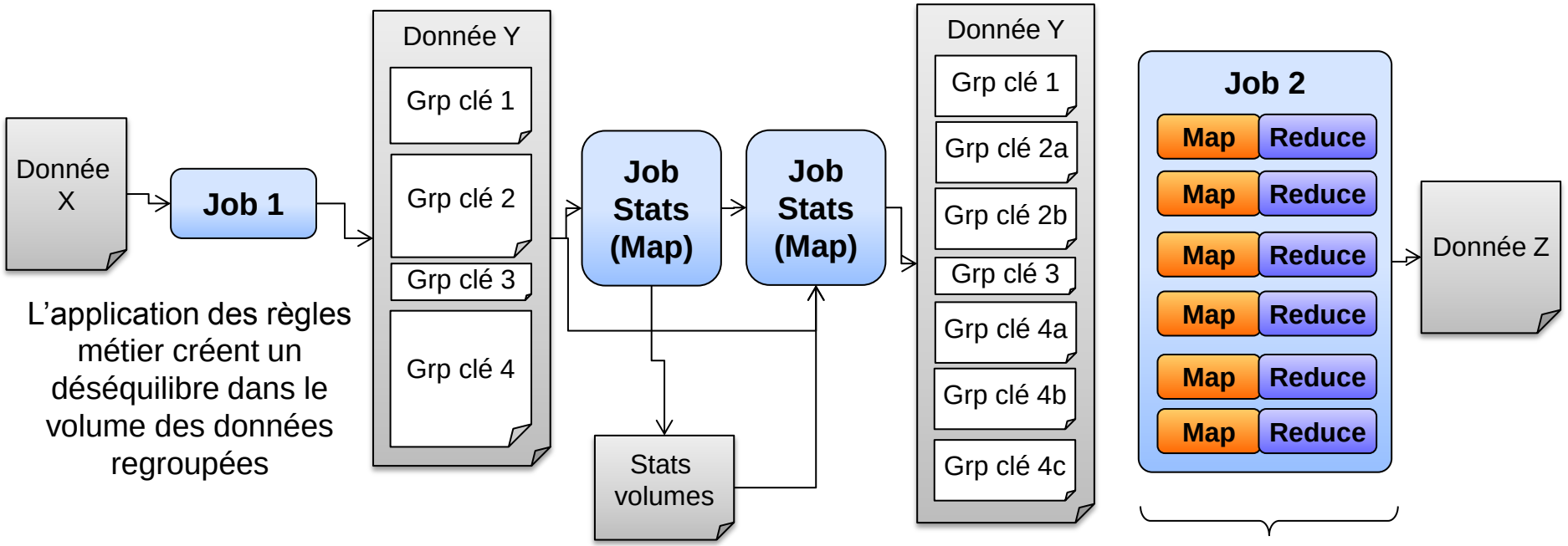
→ Risque qu'un reducer ne tienne pas en mémoire

AIRFRANCE KLM

Cas d'utilisation

Equilibrage des traitements (2/2)

KARMA



L'application des règles métier créent un déséquilibre dans le volume des données regroupées

Analyse les volumes / clé et calcul un discriminant technique pour mieux équilibrer les groupes de clés

Les temps de traitement sont équilibrés
Levée du risque lié à la mémoire nécessaire au traitement du reduce

AIRFRANCE KLM

Axes d'amélioration et évolutions

Performances et accès aux données

KARMA

- La performance
  - Paramétrage complexe de la grille Hadoop
    - Taille des blocks du HDFS
    - Nombre de slots d'exécution de Map et Reduce
  - Mise en parallèle des jobs au sein des Batchs (à l'UT et au Job)
  - Optimisation / Limitation des extractions / injections entre la DB et le HDFS
- Accès aux données
  - A des traitements Batchs d'applications tierces
  - A des traitements non batch
    - Traitements interactifs
    - Traitements évènementiels

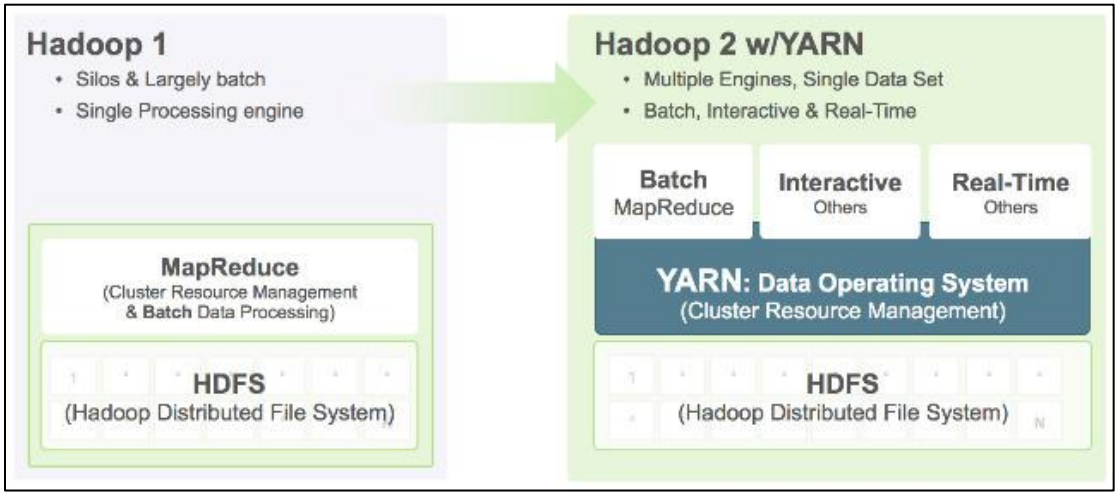
➔ Ces améliorations et nouvelles possibilités passent en partie par la migration vers Hadoop 2

Axes d'amélioration et évolutions

Migration vers Hadoop 2

KARMA

- Introduction de YARN
  - Plusieurs moteurs d'exécution
  - Meilleure gestion des ressources
  - amélioration des performances
  - Permet de gérer plusieurs applications



- Traitement interactifs
  - HIVE, HBASE, …
  - Ouverture des données du HDFS à des applications tierces
  - Accès aux données simplifié par l'utilisation de langages de Scripts / SQL Like
- Traitement temps réel :
  - Storm, Spark Streaming, …
  - Intégration des évènements / CEP

AIRFRANCE KLM

Annexes

KARMA

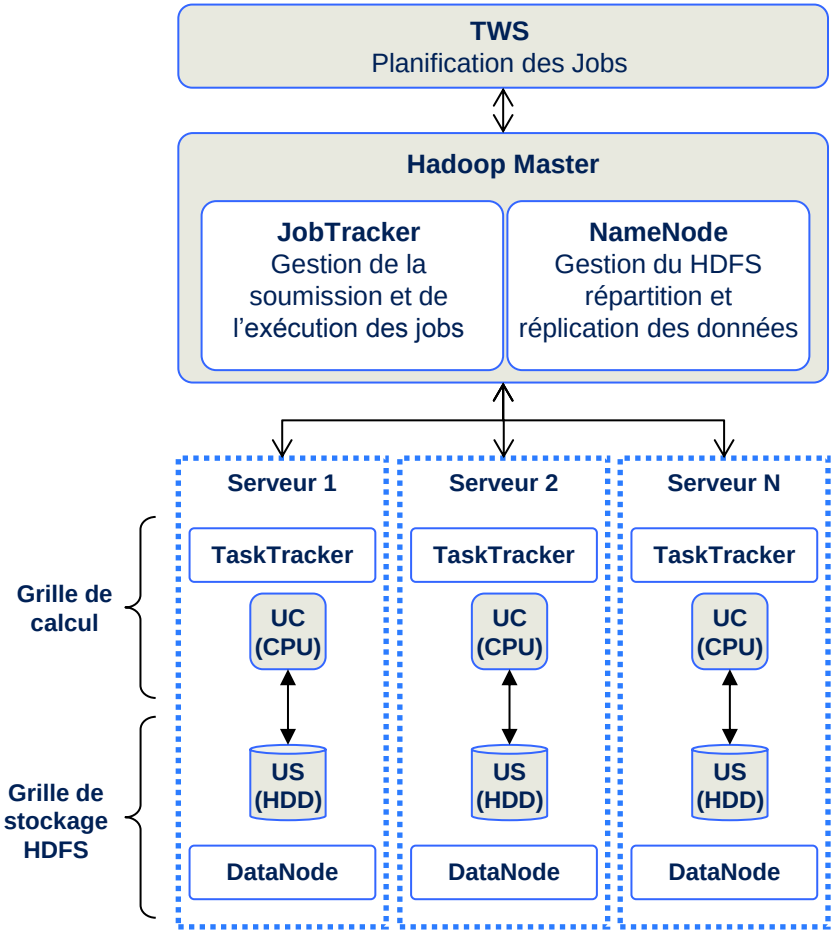
AIRFRANCE KLM

Présentation d'Hadoop

Fonctionnement général

KARMA

- Objectifs d'Hadoop
  - Paralléliser et distribuer des traitements sur une ferme de serveurs pour améliorer les performances et permettre des traitements dont les volumes de données sont (très) importants.
- Ce que fournit Hadoop
  - Moteur d'exécution
    - Gère la // et la distribution des traitements
    - Gère la distribution et la réplication des données
  - Des API de développement qui implémentent l'approche MapReduce et permet d'accéder au HDFS
  - Un système de stockage distribué le HDFS (fichiers)
    - Supporte différents formats (CSV, XML, JSON, AVRO, …)
- Des outils complémentaires
  - Supervision
  - Import / Export DB ←→ HDFS
  - Abstraction (PIG, HIVE, …)



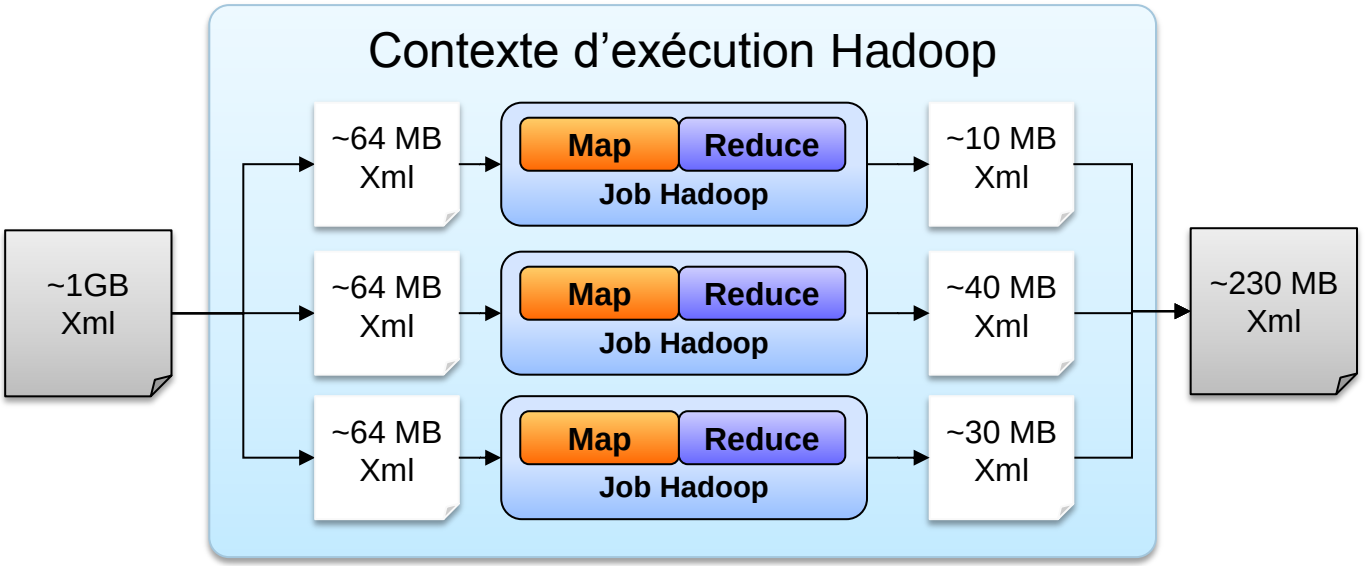
AIRFRANCE KLM

Présentation d'Hadoop

L'approche MapReduce

KARMA

- Un traitement MapReduce décompose en 2 étapes
  - L'étape de Map qui permet de lire, trier, transformer, et regrouper les données nécessaire au traitement métier. (Traitement de préparation)
  - L'étape de Reduce récupère les données préparées par le Mapper et implémente le besoin métier (Traitement Métier)
- Le framework Hadoop implémente cette approche en apportant les fonctionnalités suivantes :
  - Découpage des fichiers en blocks lors de la copie sur le HDFS
  - Distribue et réplique les blocks sur le HDFS
  - Distribue les jobs sur la grille de calcul
  - Exécute les tâche de Map et de Reduce



AIRFRANCE KLM