Big Data

GL4 (Option Management des Systèmes d'Information) - 2018

Chp3 – Traitement de Données

Batch vs. Streaming Processing

Dr. Lilia SFAXI
www.liliasfaxi.wix.com/liliasfaxi

Traitement par lot: Batch Processing

Présentation

- Moyen efficace de traiter de grands volumes de données
- Les données sont collectées, stockées, traitées, puis les résultats fournis
- Les systèmes de batch processing ont besoin de programmes différents pour l'entrée, le traitement et la génération des données
- Le traitement est réalisé sur l'ensemble des données
- Hadoop Map-Reduce est un exemple de système utilisant le traitement par lot

Traitement par lot: Batch Processing

Caractéristiques

- A accès à toutes les données
- Peut réaliser des traitements lourds et complexes
- Est en général plus concerné par le débit (nombre d'actions réalisées en une unité de temps) que par la latence (temps requis pour réaliser l'action) des différents composants du traitement
- Sa latence est calculée en minutes (voire plus)
- Cible les caractéristiques **volume** et **variété** des Big Data

Traitement par lot: Batch Processing

Inconvénients

- Il n'est pas possible d'exécuter des travaux récursifs ou itératifs de manière inhérente
- Toutes les données doivent être prêtes avant le début du job
  - N'est pas approprié pour des traitements en ligne ou temps réel
- Latence d'exécution élevée
  - Produit des résultats sur des données relativement anciennes

Traitement par lot: Batch Processing

Cas d'utilisation

- Les chèques de dépôt dans une banque sont accumulés et traités chaque jour
- Les statistiques par mois/jour/année
- Factures générées pour les cartes de crédit (en général mensuelles)

Traitement par Streaming

Caractéristiques

- Les traitements se font sur un élément ou un petit nombre de données récentes
- Le traitement est relativement simple
- Doit compléter chaque traitement en un temps proche du temps-réel
- Les traitements sont généralement indépendants
- Asynchrone: les sources de données n'interagissent pas directement avec l'unité de traitement en streaming, en attendant une réponse par exemple
- La latence de traitement est estimée en secondes

Traitement par Streaming
Inconvénients

- Pas de visibilité sur l'ensemble des données
  - Certains types de traitements ne peuvent pas être réalisés
- Complexité opérationnelle élevée
  - Plus complexes à maintenir que les traitements batch
  - Le système doit être toujours connecté, toujours prêt, avoir des temps de réponses courts, et gérer les données à l'arrivée
- Risque de perte de données

Traitement par Streaming
Cas d'Utilisation

- Recommandations en temps réel
  - Prise en compte de navigation récente, géolocalisation
  - Publicité en ligne, re-marketing
- Surveillance de larges infrastructures
- Agrégation de données financières à l'échelle d'une banque
- Internet des objets
- …

Traitement en Temps Réel

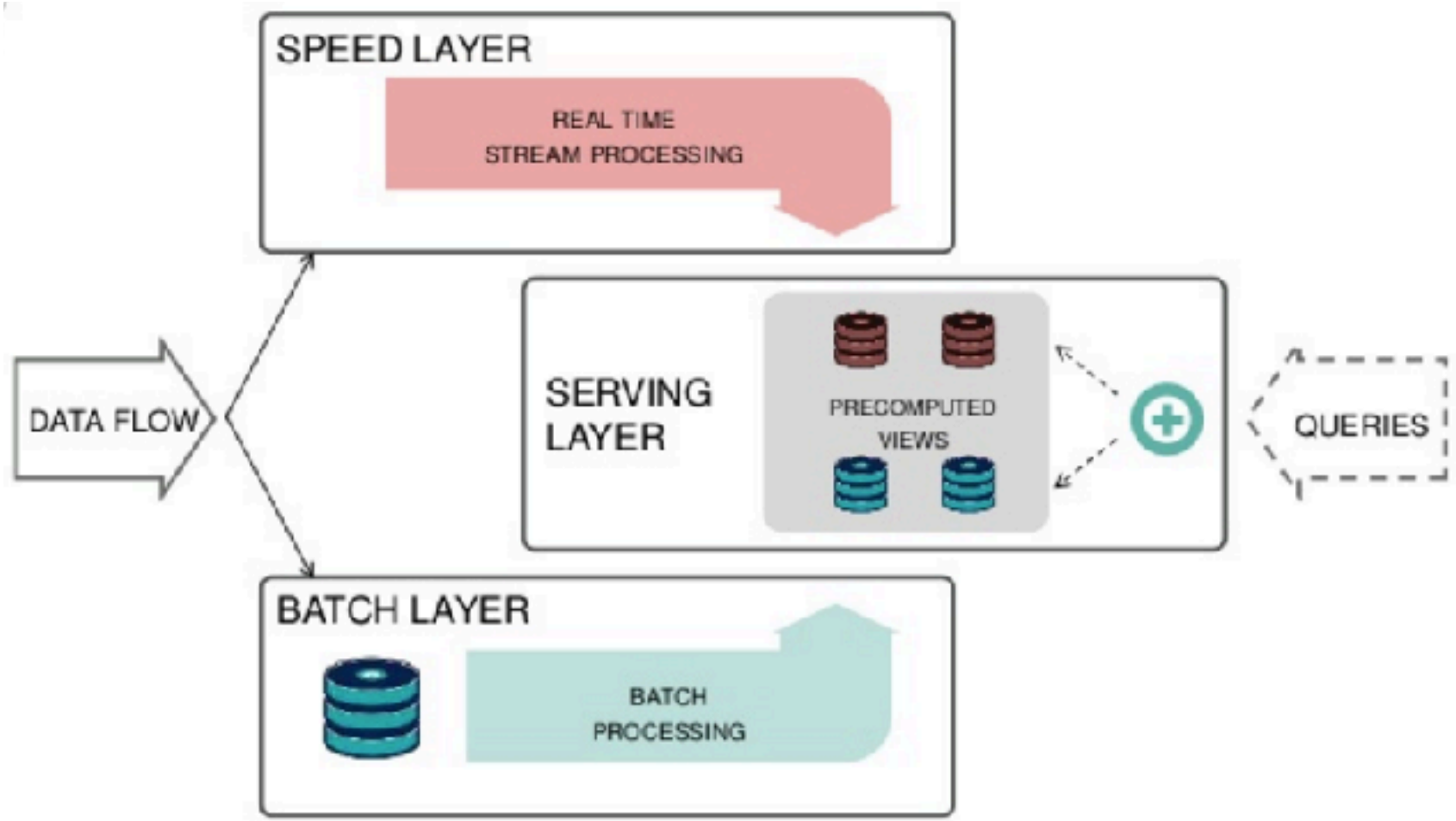
Cas d'Utilisation

- Réponse du système en quelques milli-secondes

- Plus approprié pour les APIs synchrones et interactifs

Lambda Architecture

Architecture



Lambda Architecture

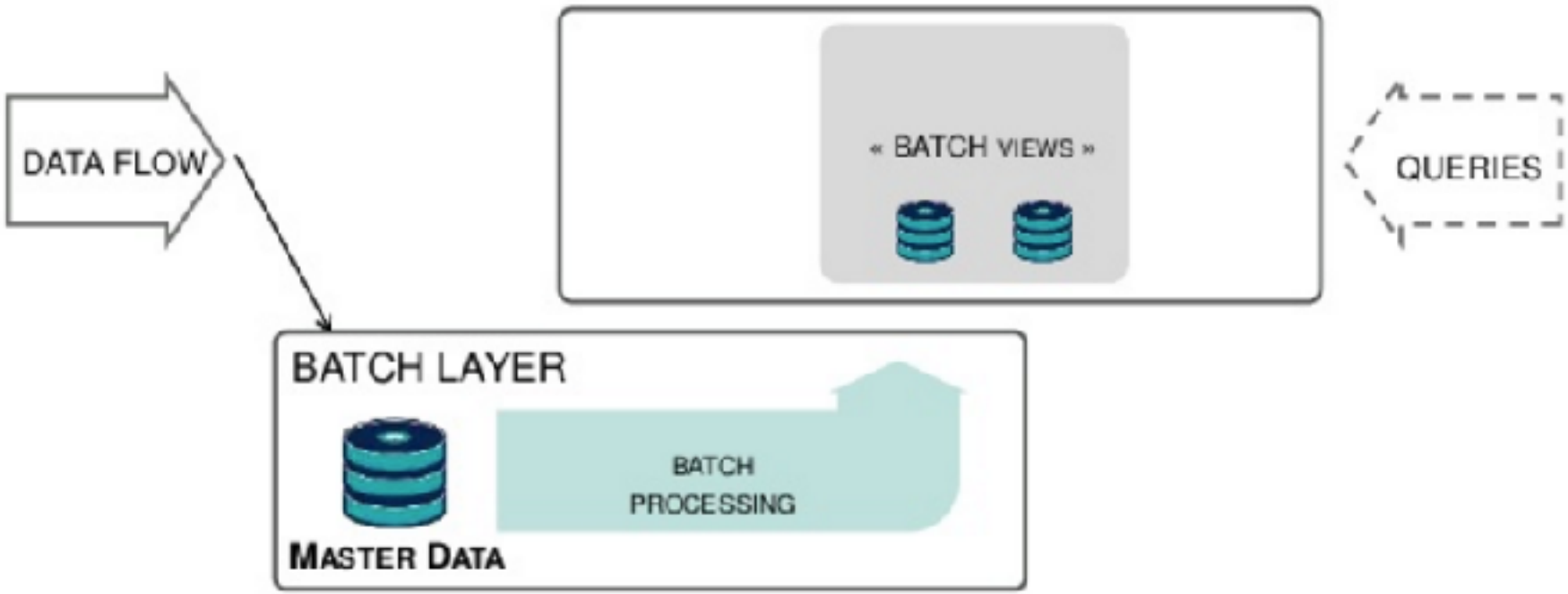
Couches

- Batch Layer
  - Gérer l'unité de stockage principale: des données brutes, immuables et complètes
  - Permet de précalculer et conserver les résultats de requêtes appelées batch views
- Serving Layer
  - Permet d'indexer les batch views pour qu'elles soient requêtées au besoin avec une faible latence
- Speed Layer
  - Permet d'accommoder toutes les requêtes sujettes à des besoins de faible latence
  - Utilisation d'algorithmes rapides et incrémentaux
  - Gère les données récentes uniquement

Lambda Architecture

Batch Layer

- Stockage maître + Traitements Batch



Lambda Architecture

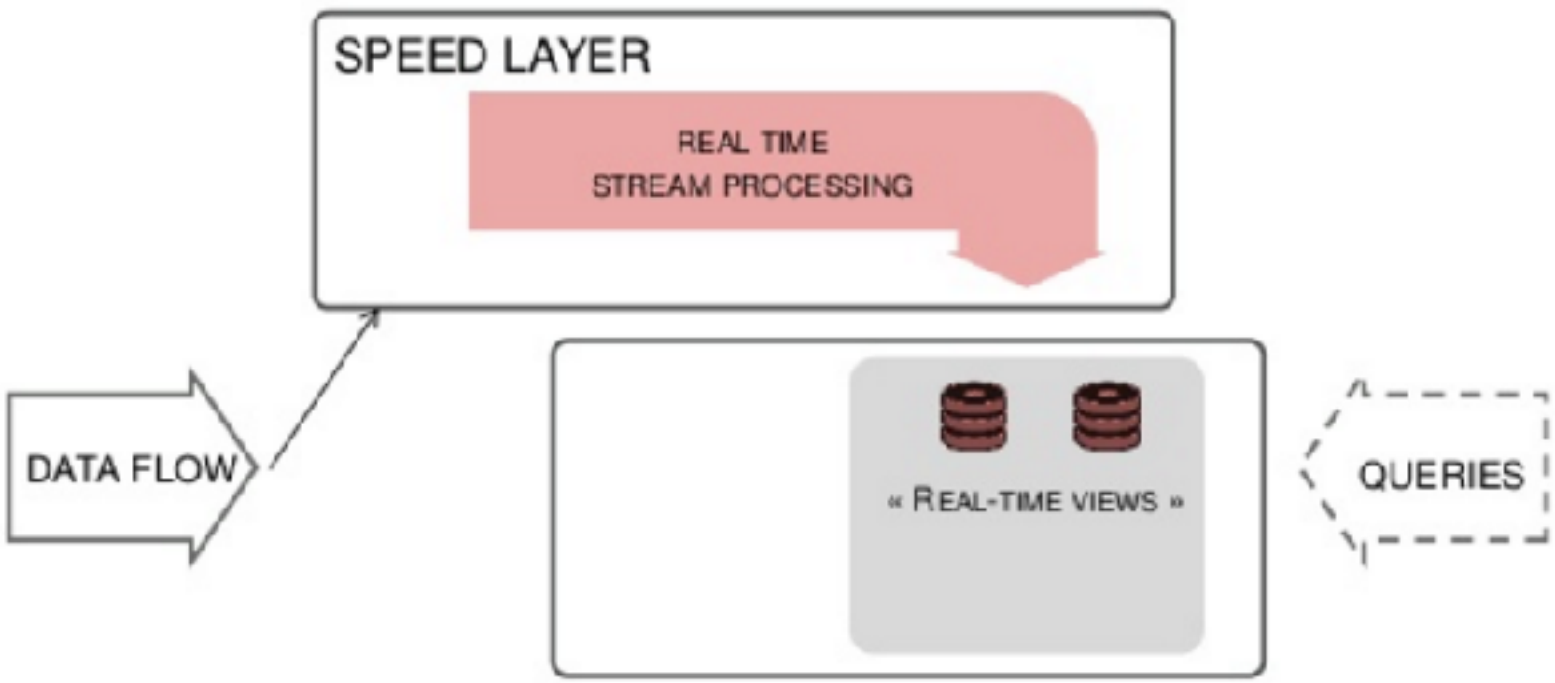
Batch Layer

- Besoins:
  - Stockage scalable
    - Haute distribution
    - Support de la charge et du volume
  - Tolérance aux pannes
    - Système de réplication et distribution des données
  - Robustesse
    - Surtout concernant les évolutions du schéma
  - Permettant tout type de traitement
- Technologies
  - Hadoop
  - Avro : Système de sérialisation des données
  - Hive
  - ...

Lambda Architecture

Speed Layer

- Mise à jour des vues en continu, de manière incrémentale
- Latence: ~10ms -> qq secondes



Lambda Architecture

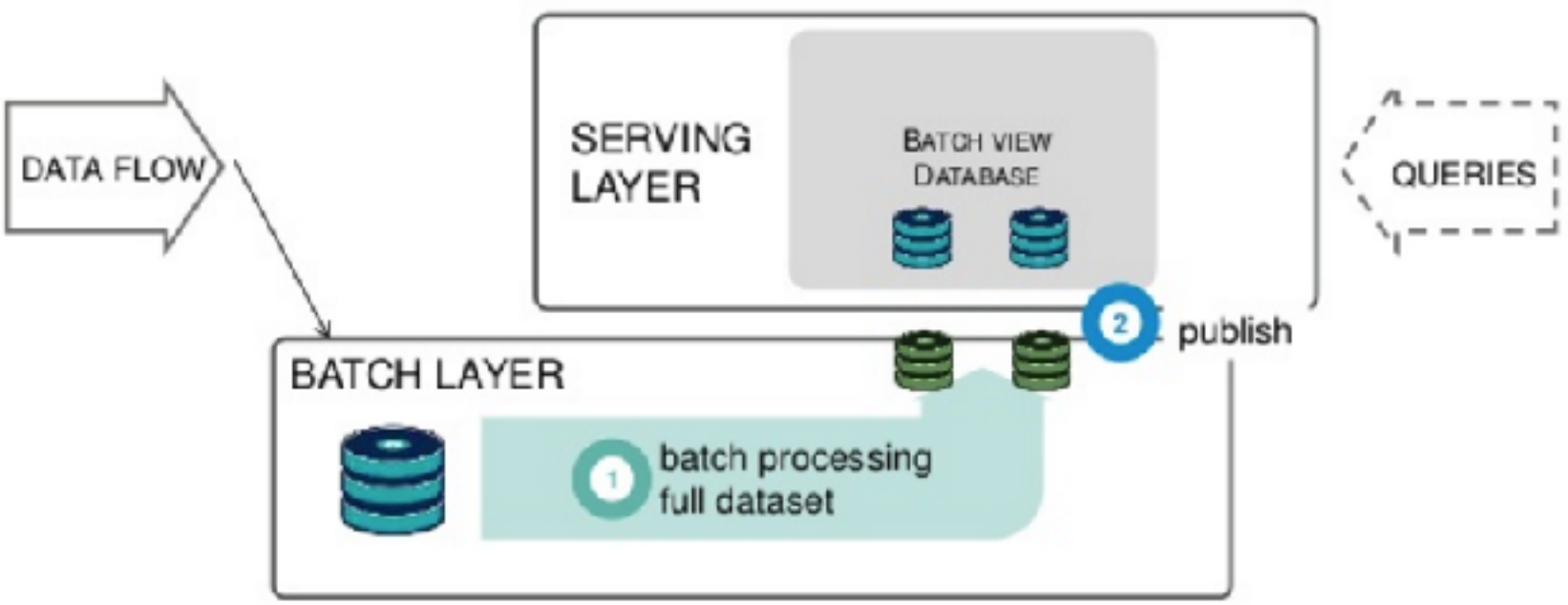
Speed Layer

- Besoins:
  - Traitement en continu (stream processing)
  - Architecture asynchrone, distribuée et scalable
  - Tolérance aux pannes
  - Garanties de traitement si possible
    - Rejeu possible des message en cas de perte d'un noeud
- Technologies
  - Kafka : collecte temps réel
  - Akka : message-driven applications
  - Storm : Stream processing
  - Spark : Micro-batch processing
  - ...

Lambda Architecture

Serving Layer : Batch Views

- Vues pré-calculées



Lambda Architecture

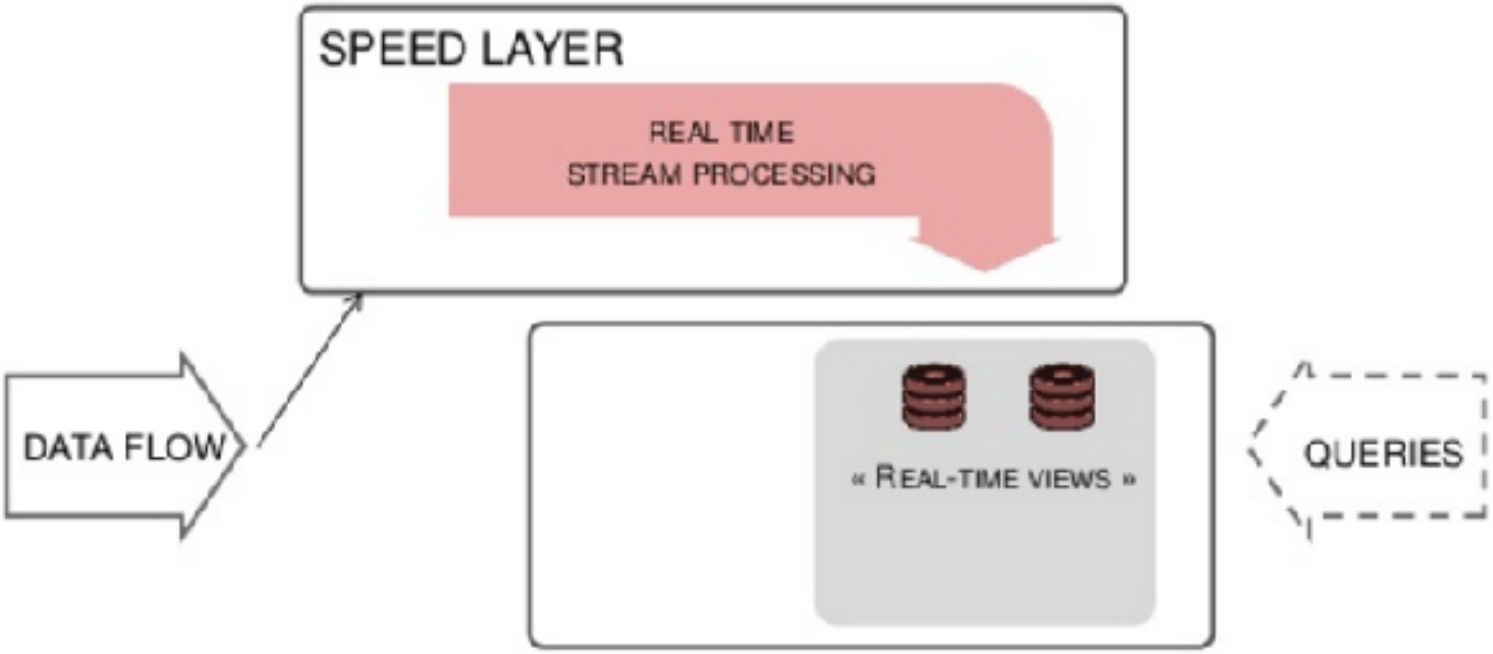
Serving Layer : Batch Views

- Besoins:
  - Écritures massives
  - Lectures indexées, accès aléatoire à faible temps de réponse
  - Scalabilité et tolérance aux pannes
- Technologies
  - Cassandra
  - Hbase
  - SploutSQL : requêtage SQL à faible latence
  - …

Lambda Architecture

Serving Layer : Real-time Views

- Vues doivent être requêtées de façon intensive et performante
- Temps de réponse court, fort débit de requête supporté



Lambda Architecture

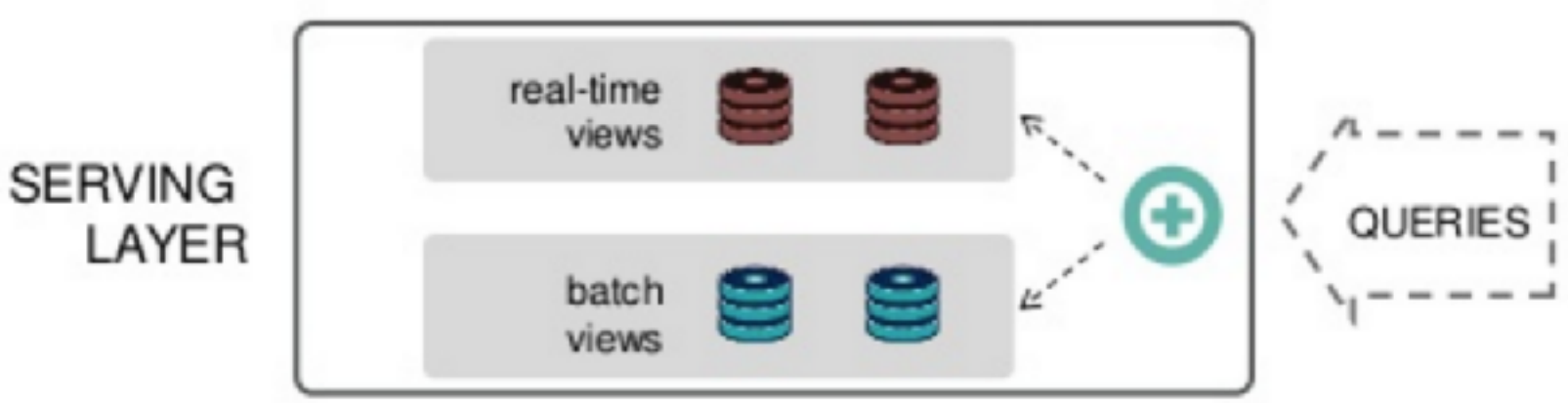
Serving Layer : Real-time Views

- Besoins:
  - Support de fortes sollicitations en lecture et écriture (mise à jour incrémentale)
  - Scalabilité et tolérance aux pannes
- Technologies
  - Cassandra
  - Hbase
  - Redis
  - ElasticSearch

Lambda Architecture

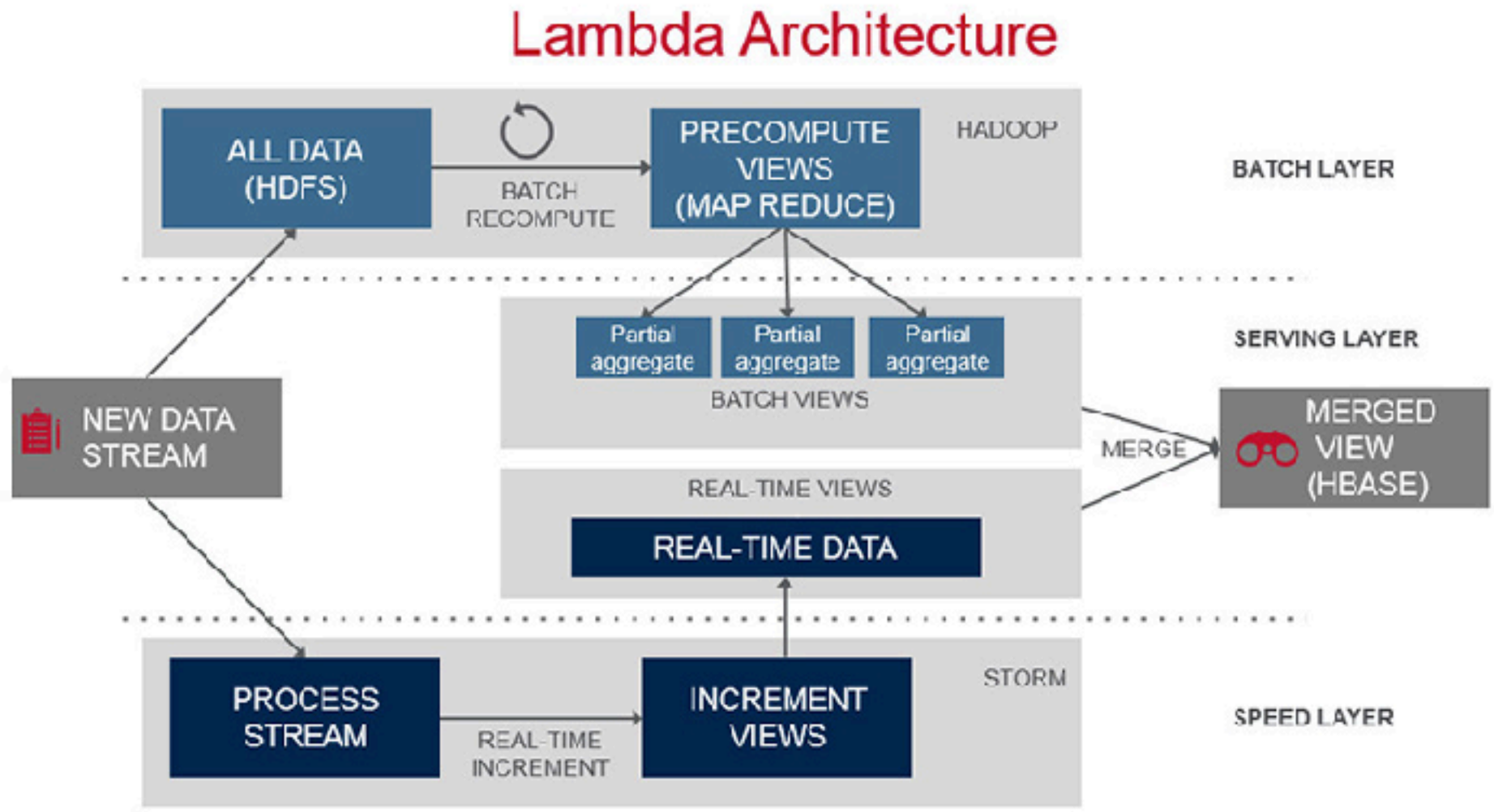
Serving Layer : Fusion

- Logique de fusion développée selon les vues et leur modélisation
- Peut s'avérer difficile
  - Expiration des vues
  - Recouvrement (intersection) possible entre les données batch et temps-réel



Lambda Architecture

Proposition d'Implémentation (MapR)



Lambda Architecture

Caractéristiques

Avantages

- Données de base sont immuables et conservées en entier
- Grande tolérance aux fautes contre les pannes matérielles et les erreurs humaines
- Supporte une variété de cas d'utilisation qui incluent un requêtage à faible latence tout autant que des mises à jour
- Capacité de « scaling out » linéaire
- Facilement extensible

Inconvénients

- Peut s'avérer coûteuse en terme de mise en place et expertise

Autres Architectures

- Encore des lettres grecques…
- Kappa :
  - http://radar.oreilly.com/2014/07/questioning-the-lambdaarchitecture.html
- Zeta :
  - http://fr.slideshare.net/MapRTechnologies/next-generationenterprise-architecture-41966097
- Iot-a :
  - http://iot-a.info/

En Conclusion…

- Tout est dans l'art d'être polyglotte…
  - Polyglot Programming
    - Utilisation de plusieurs langages et paradigmes de programmation dans une seule application
  - Polyglot Persistence
    - Utilisation de plusieurs technologies de stockage

Sources

- Présentations
  - Lambda Architecture: comment réconcilier Big Data et temps réel, Mathieu Despriee
- Articles
  - Lambda-Architecture , MapR Developer Center: https://www.mapr.com/developercentral/lambda-architecture