# L'architecture Lambda

Le but de l'architecture Lambda est de fournir un modèle de traitement presque temps réel sur des volumes importants de données, en proposant un nouveau modèle de calcul. Ce modèle essaie de trouver l'équilibre entre la tolérance aux pannes, les contraintes de latence (latence très faible pour les lectures/écritures) et le débit des disques durs en se basant à la fois sur les traitements batch qui fournissent des vues batch et les traitements temps réel qui fournissent des vues, puis les joint avant leur présentation.

Hadoop n'est pas capable de traiter un grand volume de données qui doit satisfaire une faible latence, même en ajoutant d'autres serveurs de calcul, d'où la naissance de cette architecture qui ne remet pas en question le paradigme MapReduce, mais propose une amélioration, afin de contourner les contraintes de latence de Hadoop.

L'architecture Lambda est indépendante de la technologie, et se base sur le précalcul des résultats, puis à les récupérer dans une base et les envoyant au demandeur. Elle est composée de trois couches.

Couche Batch :

Mode de fonctionnement classique des applications big data type Hadoop, cette couche est responsable de deux choses : récupérer les données et les stocker en format brut (AS-IS : pour pouvoir répondre à de nouveaux besoins métier sans impacter les données initiales) dans des puits de données (Data Lake en anglais), et lancer périodiquement des traitements sur les données, pour précalculer les résultats sous forme de vue logique. Le résultat est ensuite stocké typiquement dans des bases en lecture seule et les mises à jour remplacent les vues logiques précalculées.

Cette couche peut être implémentée à l'aide d’Apache Hadoop, MapReduce, Spark

Couche de vitesse ou temps réel (Speeding) :

Traiter les nouveaux flux de données en temps réel, sans aucun prétraitement (correction des jeux des données). Cette couche minimise la latence et fournit en temps réel des vues avec les données les plus récentes. C'est un fonctionnement en mode continu unitairement pour chaque nouvelle donnée. Les résultats (les vues temps réel) fournis par cette couche ne sont pas aussi fiables, que ceux de la couche batch.

Cette couche peut être implémentée à l'aide d’Apache Storm ou Spark Streaming.

Couche de service (Serving) :

Rend exploitables les résultats précalculés par la couche batch et la couche temps réel, pour effectuer des requêtes à la volée (ad hoc).

Cette couche peut être implémentée à l'aide des technologies NoSQL Apache HBase, Cassandra, et ElasticSearch qui permettent de merger les vues batch et les vues temps réel.

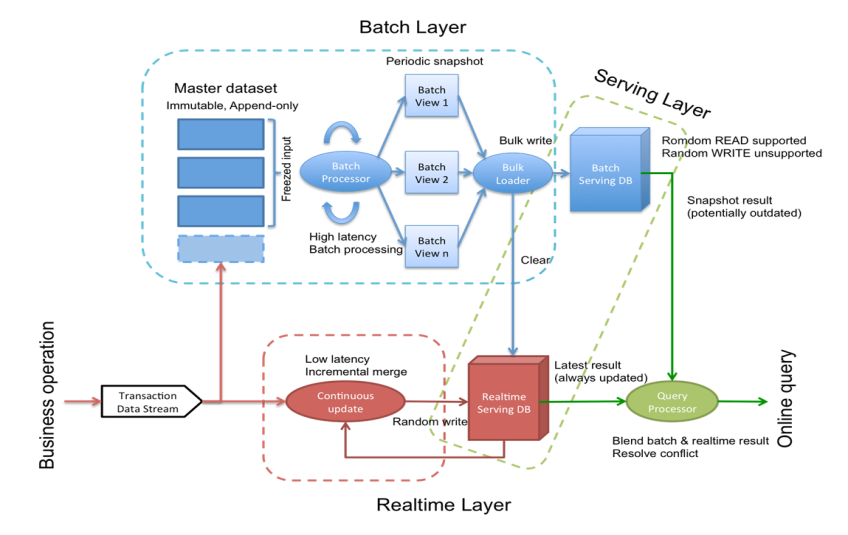


Figure 1: Architecture Lambda

Source : <http://fr.slideshare.net/MohanBavirisetty/polyglot-processing-an-introduction-10>

<https://big-data.developpez.com/tutoriels/apprendre-faire-choix-architecture-big-data/#LIV-B>

Par rapport au stockage des données l'architecture Lambda recommande de ne plus toucher à la donnée après leurs insertions dans la couche batch.

Pour plus d'informations, vous pouvez voir le site http://lambda-architecture.net/