**SPARK STREAMING**

Table des matières

[Chapitre 1 INTRODUCTION STREAMING BIG DATA 2](#_Toc105251579)

[Chapitre 2 Définition de spark streaming 2](#_Toc105251580)

[Chapitre 3 DSTREAM (DISCRETIZED STREAM) 3](#_Toc105251581)

[Chapitre 4 Transformations sur les Dstreams : 4](#_Toc105251582)

# 

# INTRODUCTION STREAMING BIG DATA

Les sources de données (capteurs, iot, time series, …etc) sont valorisées uniquement si on met en place un mécanisme (moyen) pour les capturer et les traiter. Le problème du streaming c’était toujours la latence.

Avant on pouvait faire des traitements en temps réel mais c’était extrêmement difficile :

* Modèles de programmation différents
* Double effort d’implémentation
* Double effort opérationnel.

Actuellement à coté de spark streaming il existe deux autres outils (si ce n’est plus) qui existent sur le marché pour faire du streaming à savoir :

1. Le storm : qui est l’outil qui fait réellement du streaming du streaming en temps réel mais qui n’est pas tolérant aux pannes.
2. Trident : C’est un traitement transactionnel, soit il fait tout le traitement soit il ne fait rien. Son defaut c’est la faible latence.

# Définition de spark streaming

C’est une extension de l’API spark core (intégrée à partir de la version de spark 2) pour le traitement des flux données en temps réel. On dit le traitement c’est en temps réel mais réellement spark fonctionne avec des micros batch, il fait des micro pauses pour faire les traitements.

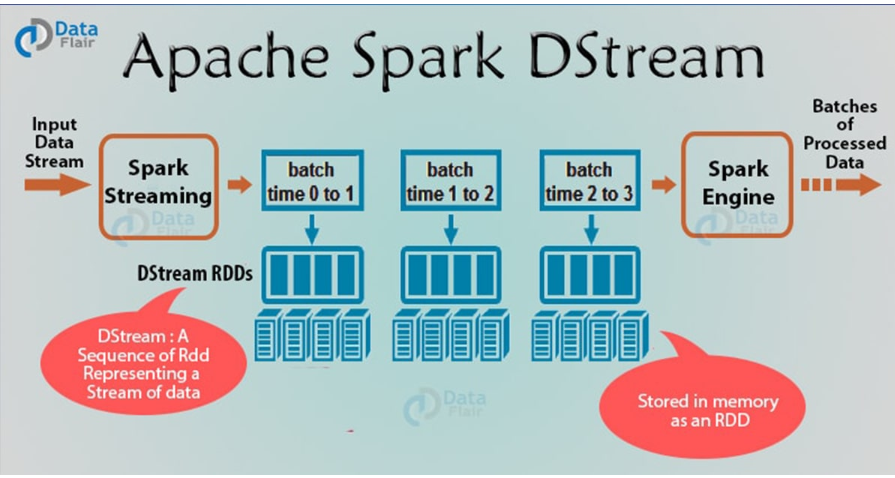
Il est scalable, à haut débit de données qu’il peut traiter, tolérant aux panes, et surtout il peut se connecter à de nombreuses sources et/ou cibles de données (kafka, S3, base de données relationnelles, flume, kafka, kenesis (aws kafka), twitter, hdfs, exposer des données sur dashboards, etc…)





# DSTREAM (DISCRETIZED STREAM)

* Spark Streaming divise le flux de données en lots appelés DStream (Discretized Stream), qui représentent une séquence des RDDs
* DStream est l'abstraction de base de Spark Streaming.
* C'est un flux continu de données, qui reçoit en entrée des données à partir de sources diverses.
* Il peut également être un flux de données généré à partir d'un flux en entrée.
* DStream est un flux continu de RDD. Chaque RDD contient des données sur un intervalle particulier.



Spark Streaming fournit deux catégories de sources de streaming intégrées :

1. Sources de base : Sources directement disponibles dans l'API StreamingContext. Exemples : systèmes de fichiers et connexions socket.
2. Sources avancées : Des sources comme Kafka, Kinesis, etc. sont disponibles via des classes utilitaires supplémentaires. Celles-ci nécessitent une liaison avec des dépendances supplémentaires, comme indiqué dans la section de liaison

# Transformations sur les Dstreams :

Semblables à celles des RDD, les transformations permettent de modifier les données du DStream d'entrée.

|  |  |
| --- | --- |
| Transformation | Description |
| map (funct) | Renvoie un nouveau DStream en passant chaque élément du DStream source via une fonction func . |
| flatMap ( fonction ) | Similaire à map, mais chaque élément d'entrée peut être mappé à 0 ou plusieurs éléments de sortie. |
| filter ( fonction ) | Renvoie un nouveau DStream en sélectionnant uniquement les enregistrements du DStream source sur lesquels func renvoie true. |
| union ( autreFlux ) | Renvoie un nouveau DStream qui contient l'union des éléments dans le DStream source et otherDStream |
|  | Renvoie un nouveau DStream de RDD à un seul élément en comptant le nombre d'éléments dans chaque RDD du DStream source. |
|  | Renvoie un nouveau DStream de RDD à un seul élément en agrégeant les éléments de chaque RDD du DStream source à l'aide d'une fonction func (qui prend deux arguments et en renvoie un). La fonction doit être associative et commutative pour pouvoir être calculée en parallèle. |
| countByValue () | Lorsqu'il est appelé sur un DStream d'éléments de type K, renvoie un nouveau DStream de paires (K, Long) où la valeur de chaque clé est sa fréquence dans chaque RDD du DStream source. |
|  |  |
|  |  |