Kafka Streams

Auteurs :
Ergi SALA : ergi.sala@etu.univ-grenoble-alpes.fr
Tom SOLVERY : tom.solvery@etu.univ-grenoble-alpes.fr

Résumé :

Kafka streams est une bibliothèque client permettant de développer des applications et microservices où les données d'entrée et de sortie sont stockées dans des clusters Kafka. Kafka streams a pour objectif de construire facilement des applications qui agiront comme des processeurs de flux de données en consommant le flux d'entrée provenant d'un ou plusieurs sujets, les traitant efficacement et produisant un flux de sortie vers un ou plusieurs sujets.

Cet outil peut être utilisé comme système de messagerie, tracker d'activité pour le web, monitoring, ou processeur de flux de données.

Abstract :

Kafka streams is a client library for developing applications and microservices where input and output data is stored in Kafka clusters.
Kafka streams aims to easily build applications that will act as data flow processors by consuming the input stream from one or more topics, effectively processing them and producing an output stream to one or more topics.

This tool can be used as messaging system, activity tracker for the web, monitoring, or data flow processor.

Synthèse :

Apache Kafka :

Kafka est une plateforme de flux distribuée.
Cet outil permet de produire et de s'abonner à des flux de données, de les stocker avec une tolérance au pannes, et de les traiter.

Il est utilisé pour concevoir des flux de données en temps réel afin de transférer de manière fiable des données entre applications ou microservice.
Il est aussi utilisé pour concevoir des applications qui transforment ou réagissent à des flux de données.

Kafka est lancé comme un cluster sur potentiellement plusieurs pouvant être éparpiller sur différent datacenters. Le cluster enregistre les flux de données par catégorie nommée *topics*. Chaque enregistrement est composé d'une clé, d'une valeur et d'une durée.

Kafka s'utilise alors à travers quatres API :

Consumer API : permet à une application de s'abonner à un ou plusieurs *topics* et d'utiliser les données provenant de l'enregistrement.

Producer API : permet à une application de produire un enregistrement assigné à un ou plusieurs *topics*.

Stream API : permet à une application de traiter les différents enregistrements provenant d'un ou plusieurs *topics* et de produire des enregistrements en les assignant à un ou plusieurs *topics*

Connector API : permet de créer et d'exécuter des producteurs et consommateurs afin de connecter des *topics* sur des applications déjà existantes comme par exemples des bases de données.

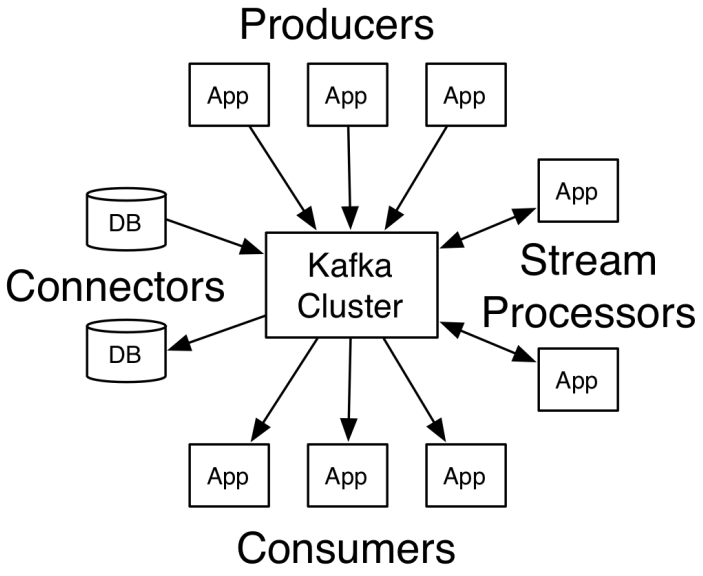

Figure 1 : Kafka platform architecture

Kafka utilise le protocole TCP pour communiquer entre le client et le serveur.

Kafka Streams :

Kafka streams (Streams API) est une bibliothèque permettant de concevoir des applications qui vont traiter différents flux de données en entrée et renvoyant différents flux en sortie. Ces données proviennent d'un ou plusieurs *topics* stockés dans le kafka cluster et crée un nouveau flux de donnée qui sera enregistré dans le Kafka Cluster.

Architecture :

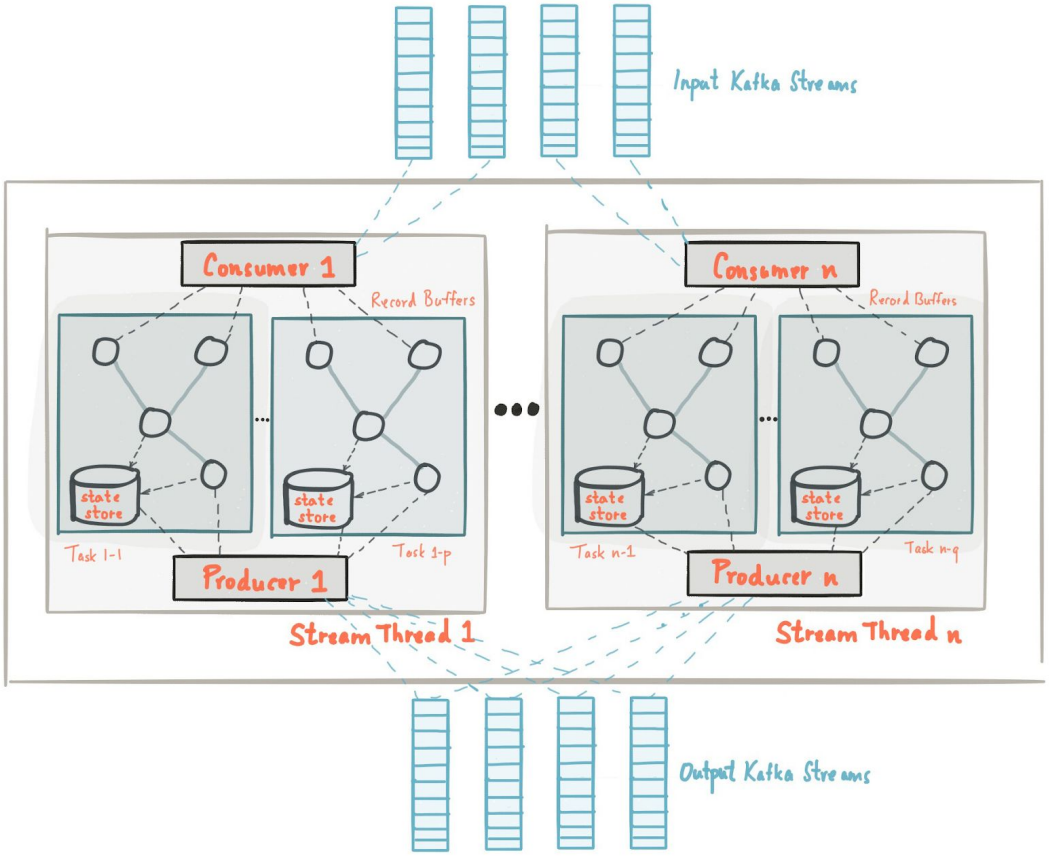

Figure 2 : Kafka stream application architecture

L'avantage d'un processeur de flux c'est qu'il peut être totalement distribué, ou lancer de façon multi-threadé ce qui permet de traiter des données très rapidement.

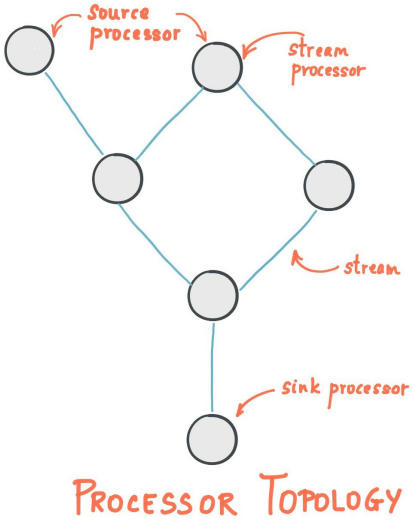

Figure 3 : Stream node topology

Une application stream processor peut être organisée avec différent noeuds. Chaque correspond à une transformation des données et sont donc reliés par des flux de données. Le premier noeud (source processor) est celui qui va récupérer le flux provenant des consommateurs et la transmettre au autres noeuds. Le dernier noeud (sink processor) est celui qui va récupérer le flux final et le transmettre au producteur.

Afin d'effectuer différentes transformations, l'API Stream propose deux classes : KStream et KTable.

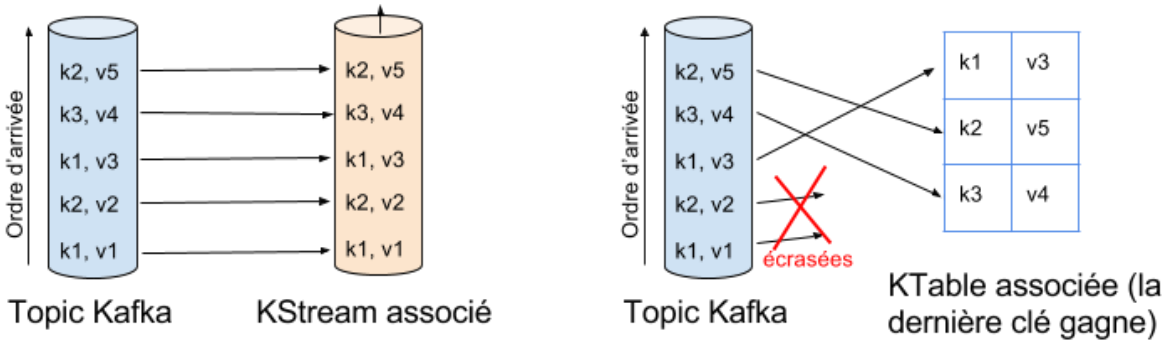

Figure 4 : KStream VS KTable

La classe KStream représente les données sous forme d'un flux continue sans interruption. Les données sont sous la forme d'une paire (clé,valeur).
La classe KTable est une représentation compacté du flux de donnée qui garde seulement la dernière valeur.

Démonstration :

WordCount : https://kafka.apache.org/23/documentation/streams/quickstart

Kafka Music Application :
https://docs.confluent.io/current/streams/kafka-streams-examples/docs/index.html

Sources :

Apache Kafka : https://kafka.apache.org/
Confluent : https://docs.confluent.io/current/kafka/introduction.html#intro-to-ak
Blog des octos :
https://blog.octo.com/kafka-streams-encore-un-framework-de-stream-processing/