COURS M2

STREAM PROCESSING

PARTIE 1: THEORIE

14.02.2018

UNIVERSITÉ PARIS-EST

Atos Worldwide IT Partner

OBJECTIFS DU COURS

- Comprendre les architectures contemporaines
- Appréhender l'écosystème temps-réel
- Mise en place d'une pipeline
  - Ingrédients essentiels
  - Cas d'usages concrets
  - Vue sur les Outils
- Demonstration
- Optimisation et analyse des goulots d'étranglement

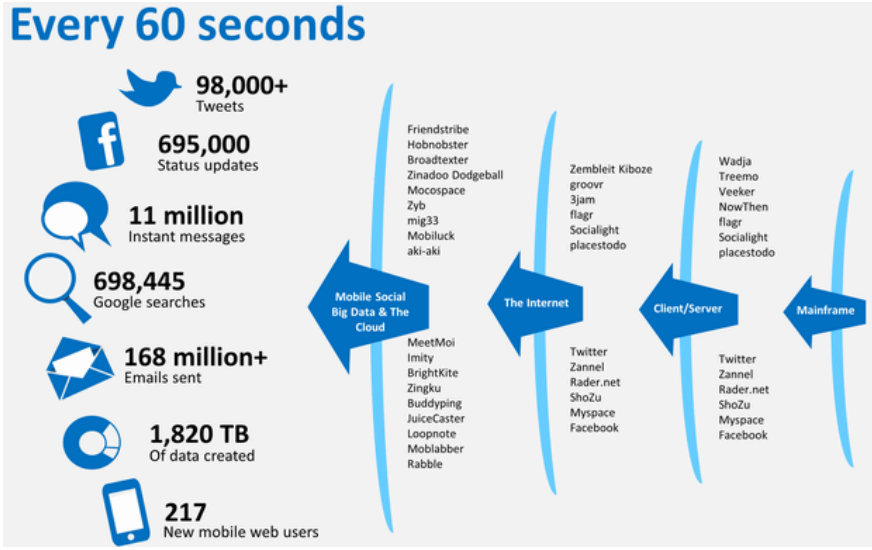
PRE-REQUIS DU COURS

- Connaître les commandes Linux de base
- Connaître les principes de la POO, des bases en Java ou Python seront nécessaires
  - Ecriture de code pouR structurer une pipeline
- Environnement Linux ou Mac sont préférables sinon windows 10.
  - Installation de Docker sur la machine
  - Ouverture compte AWS
  - JDK > 8

Stream Processing : Pourquoi ?

Peta Octets à grande vitesse…

- Taille des données ne cesse de croître
  - Réseaux sociaux, blogs, click-streams, logs…
- 2,5 trillions d'octets de données sont générés.
- 90% des données créées dans le monde l'ont été au cours des 2 dernières années.
- Prévision d'une croissance de 800% des quantités de données à traiter d'ici à 5 ans.



…a traiter en temps-réel

- **Conduite autonome**
  - informations de navigation requises
  - multiples capteurs à synchroniser
  - 1 Gb de données/min/voiture

- **Suivi et optimisation de trafic routier**
  - Dizaines de milliers d'évènements/sec
  - Dizaines de milliers de requêtes/sec

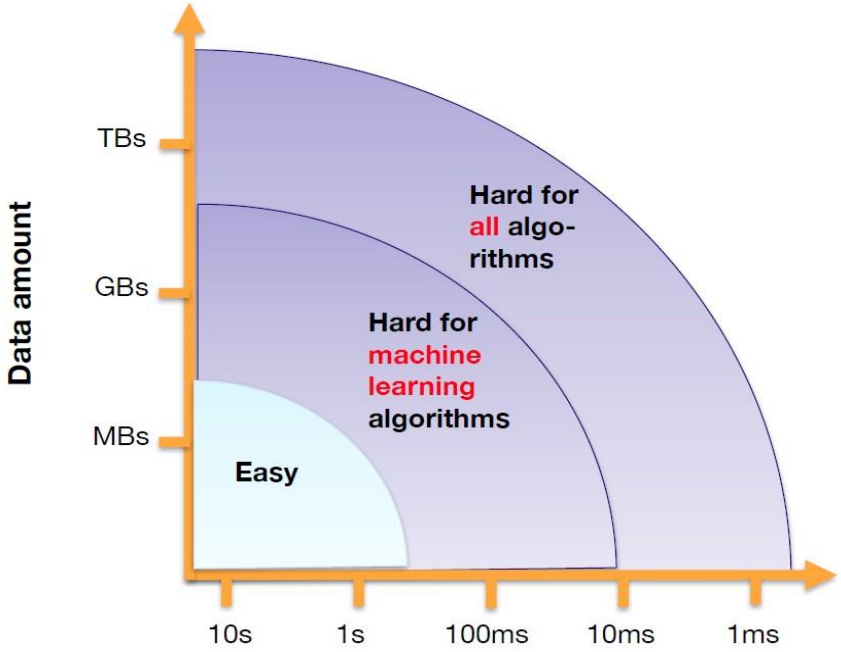
- **Pré-traitement de données de capteurs**
  - CERN → 1Pb mesures/sec
  - impossible à stocker ou traitement directement
  - Pré-traitement rapide est la solution utilisée

CERN

Complexité de la tâche

- Traitement d'énormes quantités d'événements en continu (filtrer, agréger, ordonner, automatiser, prédire, agir, surveiller, alerter)
- Réactivité en temps réel
- Performances et évolutivité
- Intégration rapide avec l'infrastructure existante
- Intégration rapide avec les sources de données en entrée et sortie
- Délai de déploiement court
- UIX et Productivité des utilisateurs
- Analytics
- Accès aux requêtes continu ad-hoc

Big Data VS Fast Processing



Stream Processing : Définitions

Streams/évènements

- Données non délimitées
  - Conceptuellement infini, éléments de données / événements en croissance continue
  - Flux de données pratiquement continu à traiter/analyser

- Modèle push
  - Production et procession de données contrôlées par la source
  - Modèle « publish/subscribe »

- Concept de temps
  - Besoin de raisonner sur quand les données sont produites et quand les données traitées doivent être produites
  - Différencier entre temps de traitement, temps d'ingestion, temps d'événement

Les évènements doivent-ils être ordonnés ?

Modèles de streams

- De manière formelle, un flux de données est une paire ordonnée (S, T)
  - S est une séquence de tuples et
  - T est une séquence d'intervalles en temps réel positifs
  - objet de données = vecteur d'attributs multidimensionnel dans un espace d'attributs continu, catégoriel ou mixte

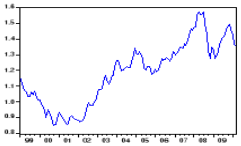
S = Si, Si+1, …
Si = <data item, timestamp>

- **Modèle « tourniquet »**
  - Eléments peuvent « entrer » et « sortir»
  - Le modèle sous-jacent est un vecteur d'éléments
  - Si est une mise à jour (incrément ou décrément) d'un élément vectoriel
  - Modèle de base de données traditionnel
  - Modèle flexible pour les algorithmes

Quel modèle est utilisé aujourd'hui par les plateformes célébres ?

- **Modèle « caisse enregistreuse »**
  - Similaire au modèle tourniquet mais les éléments ne peuvent « sortir »

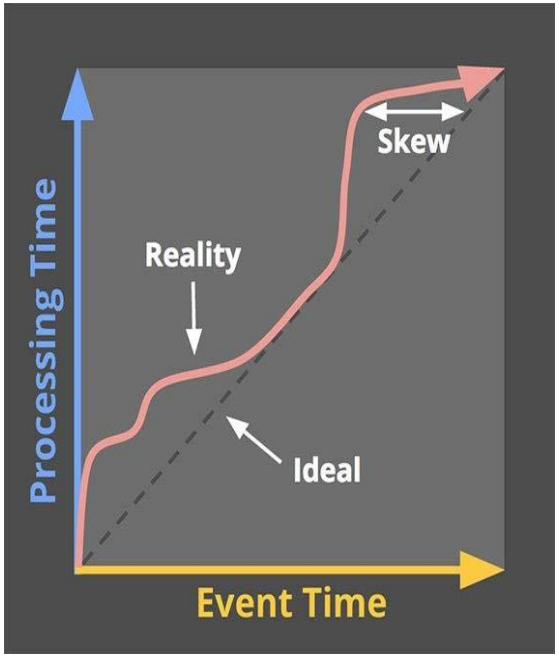
- **Modèle « séries temporelles »**
  - Si est une nouvelle entrée de vecteur
  - Le vecteur augmente



Concepts de temps

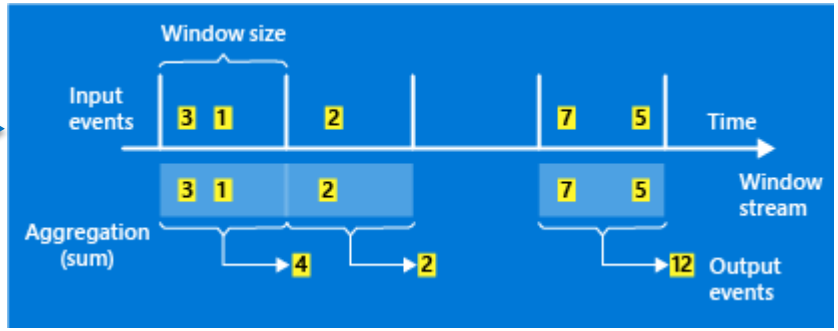
- **« Temps évènement »**
  - Quand l'évènement a-t-il été produit ?
- **« Temps Ingestion »**
  - Quand le système a-t-il reçu l'évènement ?
- **« Temps Traitement »**
  - Quand le système a-t-il traité l'évènement ?

Ces trois temps peuvent-ils coincider ?



Fenêtres de temps

- Une fenêtre contient des données d'événement horodatés permettant d'effectuer différentes opérations sur les événements de cette fenêtre.



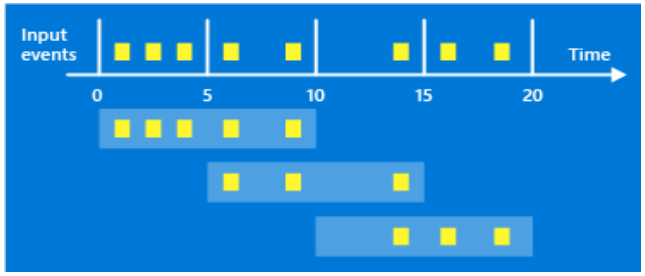
- Chaque opération de fenêtre génère un événement à la fin de la fenêtre.

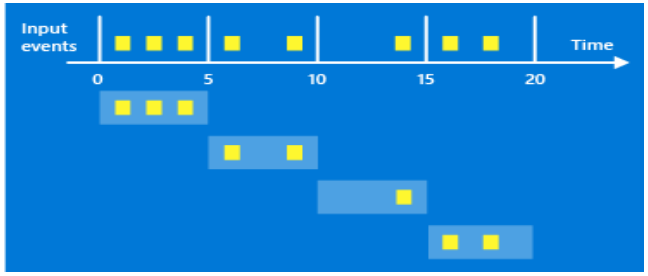
- La sortie de la fenêtre sera un événement unique basé sur la fonction d'agrégation utilisée avec un horodatage égal à l'heure de fin de la fenêtre.

Types de Fenêtres

- **« Tumbling Window »:** série d'intervalles de temps de taille fixe, sans chevauchement et contigus.

A quel schéma appartient chaque définition ?

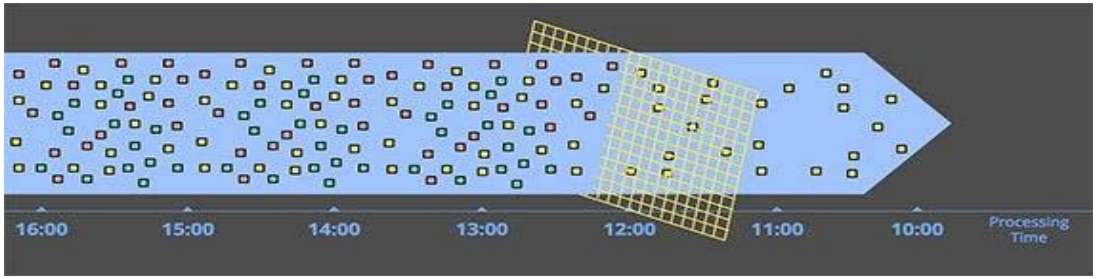




- **«Sliding Window »:** composée d'une unité de temps, de taille de fenêtre et de taille de saut(par combien chaque fenêtre avance par rapport à la précédente).

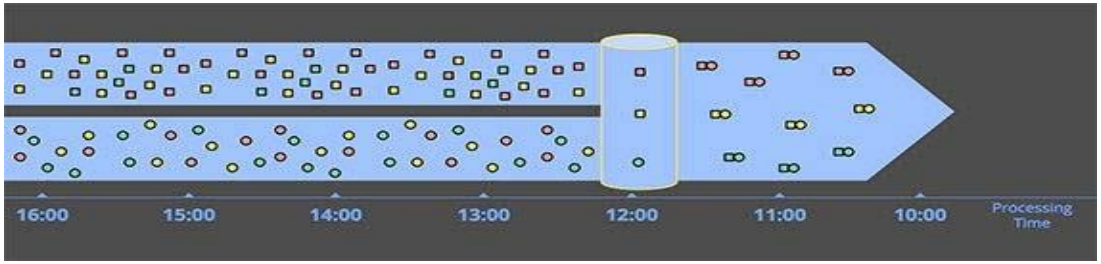
Types d'opérations

Quelle operation/état pour chaque serie temporelle de tuples ?



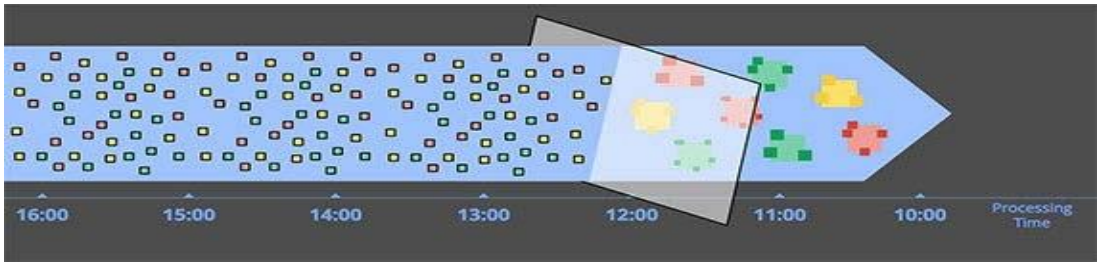
Jointure Interne

stateful



Clustering

stateless



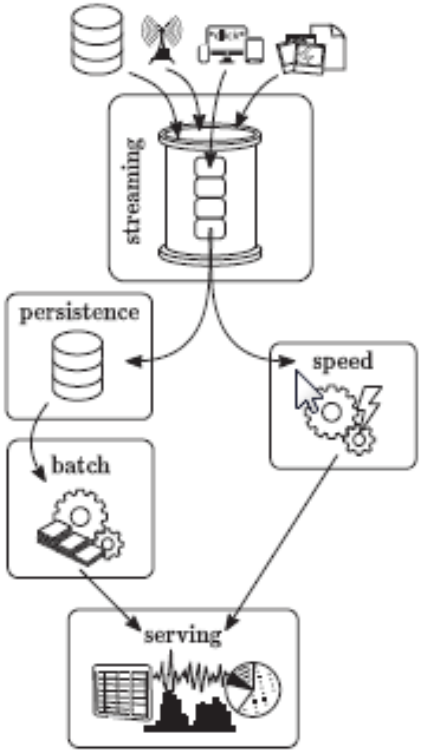
Filtrage

stateful

Stream Processing : Architectures

Types d'architectures

LAMBDA



KAPPA