C'est un sujet particulièrement important dans le contexte des villes intelligentes où de nombreuses applications traitent des flux de données…

cc BY NC ND

Deux types de traitement de données

- Traitement de données en mode batch
  - Gros volumes de données stockées sur disque
  - Fouille de données, analyse statistique
- **Traitement de flux de données en (presque) temps réel**
  - Données des smartphones et des capteurs
  - Analyse de comportement, de tendances, surveillance, notifications, guidage

2

…issus, par exemple, de capteurs en divers lieux de la ville ou des smartphones des personnes présentes dans la ville.
Des applications typiques sont, par exemple, l'analyse de tendances, l'analyse de comportements, la surveillance, les systèmes de notification et de guidage

cc BY NC ND

Défis du traitement de flux de données

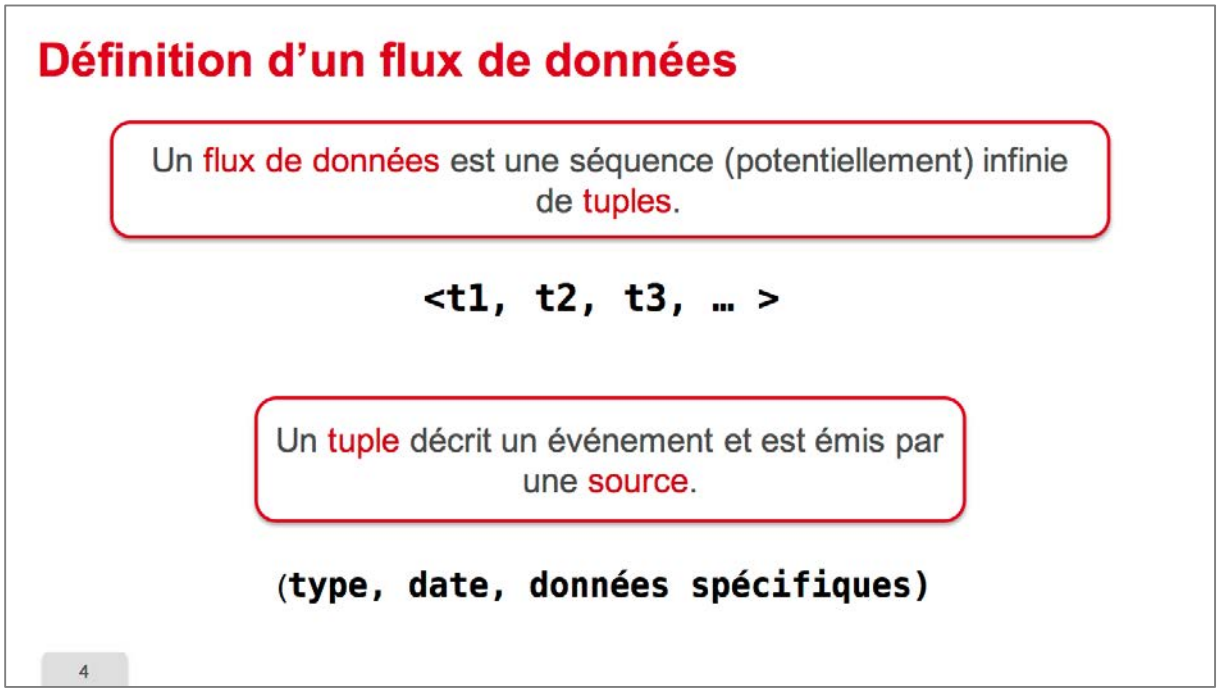
- Caractéristiques des flux de données
  - Volume
  - Variabilité
  - Vélocité

Obtenir de l'information utile en (presque) temps-réel à partir de flux de données continus

3

Rappelons, tout d'abord, quelques caractéristiques importantes des flux de données: d'une part, le **volume** de données produites, d'autre part, la **vélocité**, c'est-à-dire la vitesse à laquelle les données sont produites, et enfin la **variabilité**, le rythme de production des données et variabilité parfois dans les données produites.
L'objectif des applications de traitement de flux de données, comme par exemple l'application Sound City citée au cours de la première semaine, **produire de l'information utile à partir d'un flux de données, en presque temps réel**.

cc BY NC ND

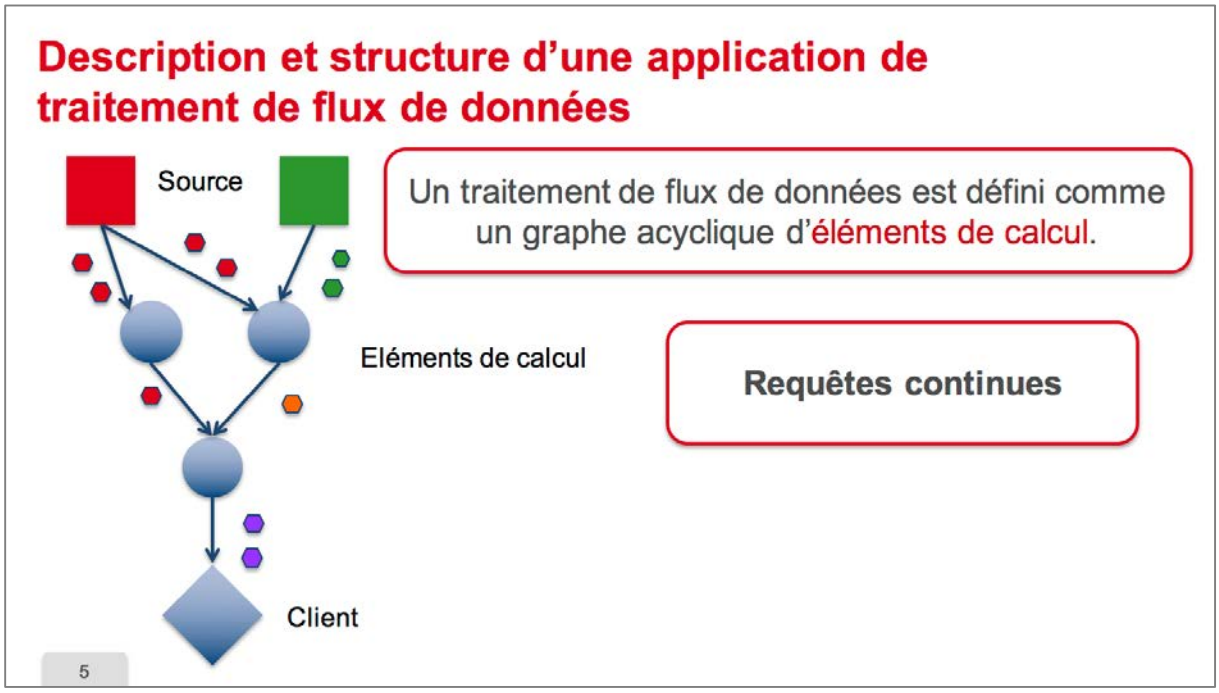


De manière plus formelle, un flux de données est une séquence potentiellement infinie de **tuples**.

Un tuple décrit un événement qui est émis par une source.

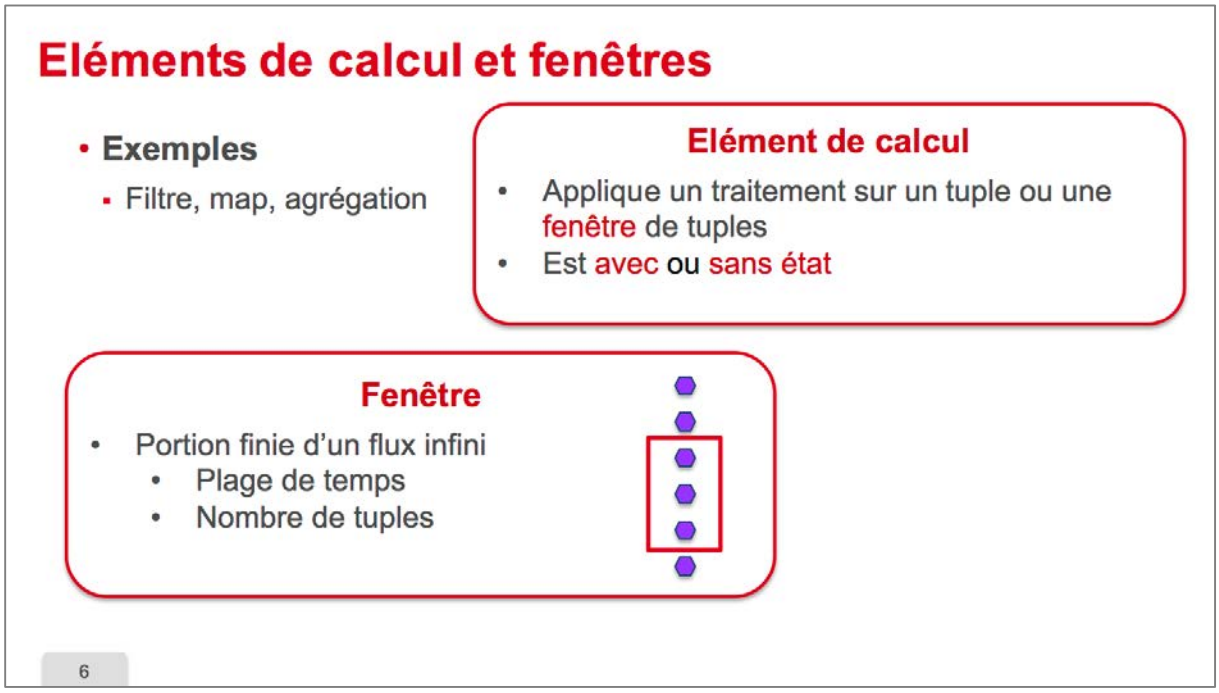
Par exemple, une caméra de surveillance émet un flux d'images de la zone surveillée. Un événement comporte un type, une date et des données spécifiques à l'événement.

CC BY NC ND



Une application de traitement de flux de données peut être représentée par un graphe acyclique d'éléments de calcul que l'on appelle également opérateurs.
Les éléments de calcul, représentés par des disques sur la figure, reçoivent les données d'une ou plusieurs sources et d'un ou plusieurs autres éléments de calcul.
A l'extrémité du graphe, le client, représenté par un losange, représente la requête continue de l'application.

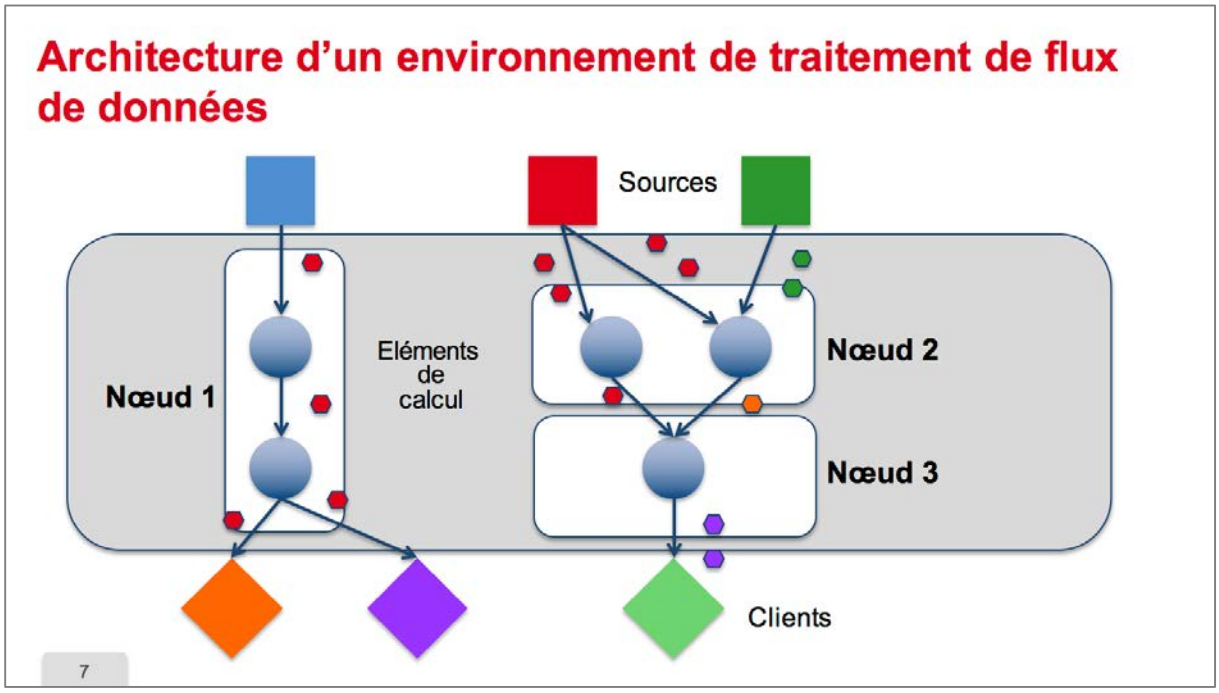
cc BY NC ND



Les éléments de calcul appliquent un traitement sur un tuple ou une **fenêtre de tuples**.
On définit une **fenêtre de tuples** comme une **portion finie d'un flux infini**. Cette portion peut être définie de différentes manières, par exemple, par une **plage de temps** ou par un **nombre de tuples** à traiter.
Les **éléments de calcul** peuvent être **avec** ou **sans état**. Et ces éléments de calcul peuvent exécuter différents types de fonction :

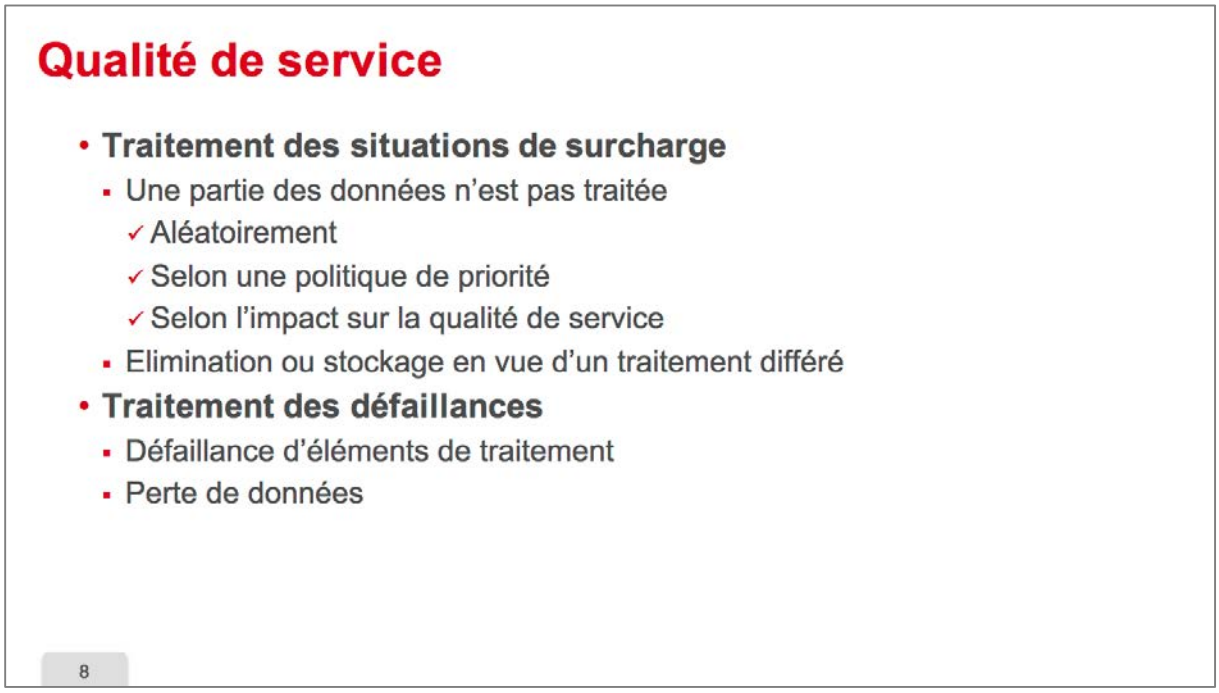
- fonctions de filtrage des données, pour éliminer certaines données,
- fonctions MAP, qui appliquent un traitement sur les données reçues, pour générer de nouvelles données en sortie,
- des fonctions d'agrégation de données.

cc BY NC ND



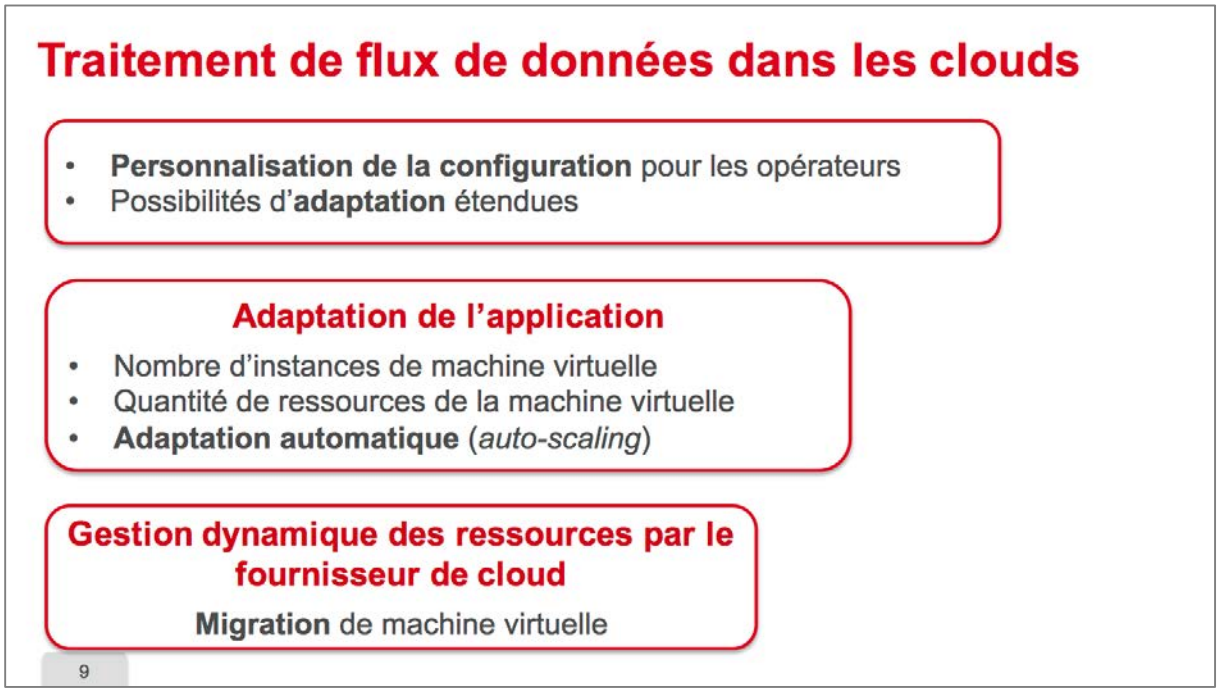
Les environnements d'exécution de flux de données assurent un traitement efficace, en quasi temps réel, des données qui arrivent en continu. Donc, ils ont en charge l'allocation des ressources pour les différents opérateurs. Donc, il s'agit, en particulier, de placer judicieusement les opérateurs sur les nœuds de calcul pour que chaque opérateur dispose des ressources dont il a besoin.

cc BY NC ND



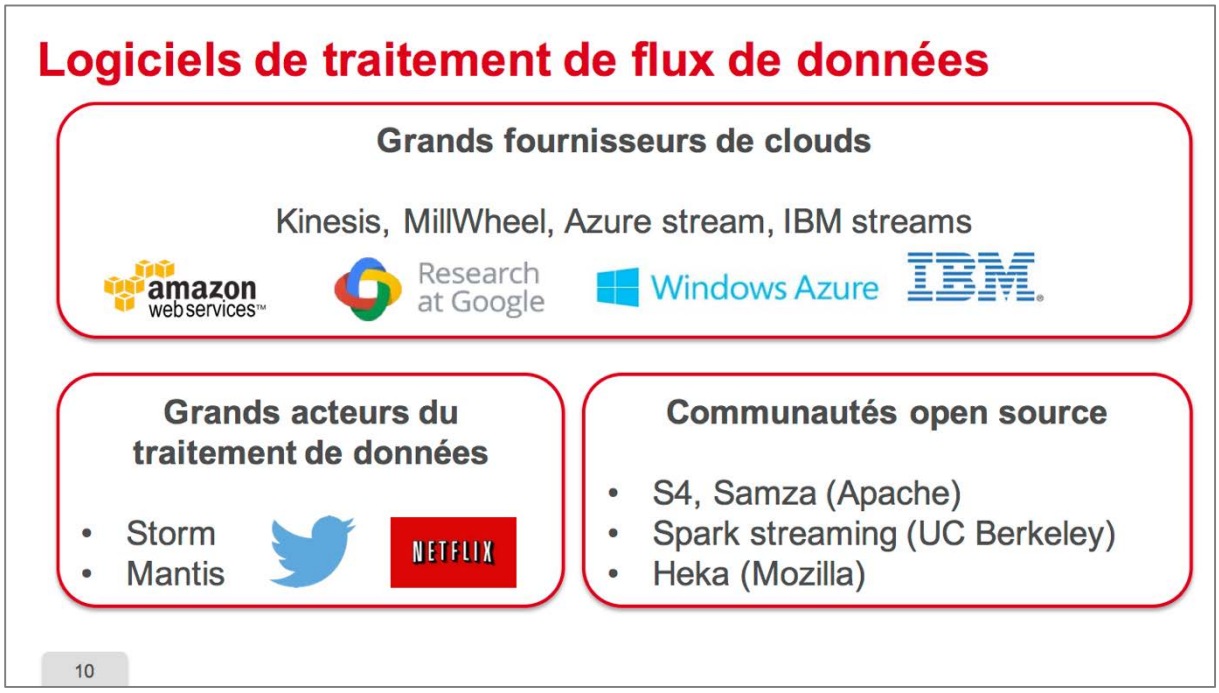
Les environnements d'exécution doivent **gérer les situations de surcharge**.
On a vu que le rythme de production des données peut varier au fil du temps. En cas de surcharge, l'environnement n'est plus en mesure de traiter toutes les données qui arrivent. Et dans ce cas-là, il applique une politique pour sélectionner les données à traiter. Ce peut être un choix aléatoire, ce peut être un choix en fonction de priorités ou un choix qui va faire en sorte de dégrader le moins possible la qualité de service.
Les données qui ne sont pas traitées peuvent être soit totalement éliminées ou bien rangées sur un disque en vue d'un traitement ultérieur une fois que le burst est terminé.
Un autre aspect important dans les environnements de traitement de flux de données est le **traitement des défaillances** qui peuvent affecter les nœuds ou bien le réseau et, surtout, qui peuvent entraîner une perte des données.
Il serait trop long de rentrer dans les détails. Je vais me borner à rappeler que toute technique de tolérance aux fautes implique la capacité à détecter les défaillances et repose sur de la redondance.

CC BY NC ND



Le cloud est une plate-forme de choix pour exécuter les environnements de traitement de flux de données.
Les éléments de calcul sont exécutés au sein de machines virtuelles.
La capacité, et la configuration, de ces machines virtuelles peut être adaptée aux besoins spécifiques des différents opérateurs.
En outre, le cloud apporte des **possibilités étendues d'adaptation**, ce qui est particulièrement intéressant pour gérer les situations de surcharge, de pics de charge.
La configuration peut être adaptée de différentes manières, par exemple, en jouant sur le nombre de machines virtuelles, en jouant sur la capacité des machines virtuelles. Et enfin, il peut être adjoint un mécanisme d'auto-adaptation de l'environnement.
Dans le contexte des villes intelligentes, un data center est susceptible d'héberger de très nombreuses applications de traitement de flux de données. Et donc, de très nombreux flux de données vont être traités en parallèle.
Du côté du fournisseur de cloud, il est essentiel de mettre en œuvre des politiques de gestion de ressources adaptées à ce type de workload dont les besoins en ressources peuvent être extrêmement fluctuants. Donc, des migrations de machines virtuelles pourront être effectuées, mais de manière judicieuse, de sorte à garantir la qualité de service attendue par les applications.

CC BY NC ND

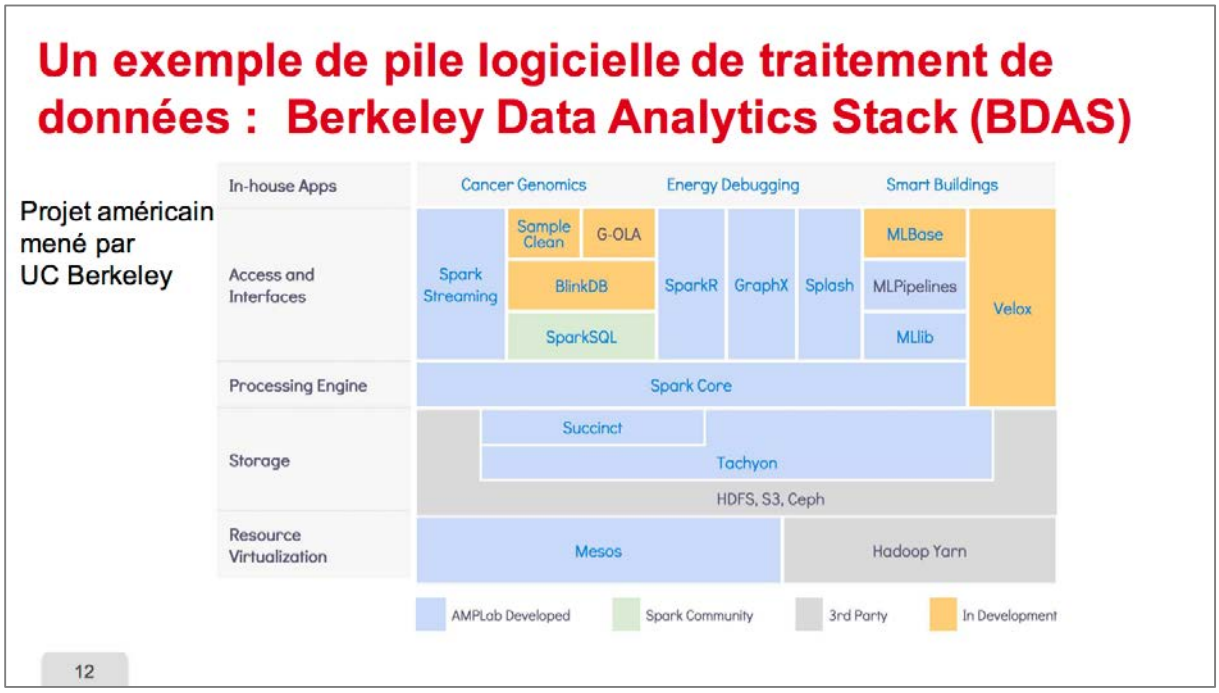


Plusieurs logiciels de traitement de flux de données existent qui émanent des principaux fournisseurs de cloud comme, par exemple, Azure Stream chez Microsoft.
Ils peuvent émaner également des grandes entreprises gérant beaucoup de flux de données comme, par exemple, l'environnement Storm, développée par Twitter.
Il existe aussi bon nombre de solutions développées en open source comme, par exemple, Spark Streaming, développée à UC Berkeley.

CC BY NC ND

Le traitement de flux de données dans le cloud donne lieu à de nombreux travaux de recherche en cours. Plusieurs directions sont explorées.
Quel service de streaming fournir dans les plates-formes - ***Stream (Data) processing as a Service*** (DaaS) ?
Comment garantir des termes de **contrats de service** pour ces services de streaming ?
Une autre direction de recherche est **l'optimisation des ressources dans les clouds** IAAS en présence de très nombreuses requêtes et de flux au comportement imprévisible.
Pour les administrateurs d'applications de traitement de flux de données se pose la question d'**optimiser le coût économique** des traitements de flux lorsque des clouds commerciaux sont utilisés.
Enfin, une autre voie de recherche intéressante est la conception des environnements de **traitement combinés** pour traiter à la fois des gros volumes de données stockées sur disques et des flux de données.

CC BY NC ND

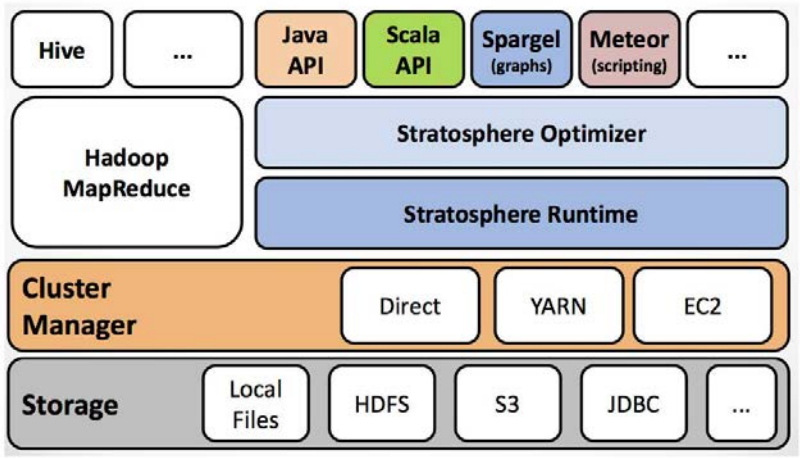


Les besoins en traitement de données et de flux de données sont considérables dans de nombreux domaines dont, évidemment, celui de la ville intelligente.
Des piles logiciels diffusées en open source ont été développées de part et d'autre de l'Atlantique pour répondre aux besoins variés en la matière.
Deux piles notables qu'il est intéressant de mentionner et qui pourraient être utilisées dans le cadre des villes intelligentes sont :

- la pile BDAS (Berkeley Data Analytics Stack), développée à l'Université de Californie, à Berkeley, qui est d'ailleurs utilisée en production

CC BY NC ND

Un exemple de pile logicielle de traitement de données : Stratosphere



Projet européen mené par TU Berlin, Humboldt University et Hasso Plattner Institute

13

- La pile logiciel Stratosphère, développée dans le contexte de l'Espace Européen de la Recherche à l'initiative de partenaires allemands.

cc BY NC ND

Un grand défi du cloud pour la ville intelligente

Traitement de données massives en temps-réel
dans le contexte de l'**Internet des objets**
pour fournir efficacement des **informations pertinentes**
à de très **nombreux utilisateurs mobiles**
en fonction de **leur localisation** et du **contexte**

14

Pour conclure la semaine, je dirais que le **traitement de données massives en temps réel**, dans le contexte de **l'Internet des objets**, pour fournir efficacement des **informations pertinentes** à de très **nombreux utilisateurs mobiles**, en fonction de leur **localisation** et du **contexte**, est le grand défi du cloud pour les villes intelligentes.

CC BY NC ND

Illustrations & photos : crédits

p. 2 : © Julien Eichinger, Fotolia ; By Camelia.boban CC BY-SA 3.0, via Wikimedia Commons ; Domaine public, Pixabay.

p. 3 : Domaine public, Pixabay.

p. 13 : droits réservés, projet Berkeley Data Analytics Stack, https://amplab.cs.berkeley.edu/software/

p. 14 : droits réservés, projet européen Stratosphere piloté par TU Berlin, Humboldt University et Hasso Plattner Institute

p.15 : © weedezign, Shutterstock

CC BY NC ND