

L'architecture en flot de données

Table des matières

I. Qu'est-ce qu'une architecture en flot de données ?	3
II. Exercice : Quiz	7
III. Comment appliquer une architecture de données ?	8
IV. Exercice : Quiz	11
V. Essentiel	12
VI. Auto-évaluation	13
A. Exercice	13
B. Test	13
Solutions des exercices	14

I. Qu'est-ce qu'une architecture en flot de données ?

Durée : 1 h

Prérequis : aucun

Environnement de travail : draw.io

Contexte

Dans le monde de l'informatique, une architecture logicielle schématise et décrit de manière symbolique les divers éléments d'un ou parfois de plusieurs systèmes informatiques, mais aussi leurs interactions ainsi que leurs interrelations. C'est lors de la phase de conception que le modèle d'architecture est produit, son rôle n'est pas de décrire ce que doit réaliser le système informatique mais plus la manière dont le système informatique doit être conçu afin de répondre au cahier des charges. Pour résumer, l'architecture ne doit pas répondre à la question « *quoi faire ?* » mais à « *Comment le faire ?* ».

Mais toutes les architectures logicielles ne se ressemblent pas. En effet, il existe plein de modèles d'architectures différents et parmi eux, il serait compliqué de tout structurer dans un seul et même type de schéma. En plus de cela, un seul et même système informatique peut, selon son niveau de granularité (c'est-à-dire selon ses différentes tailles de composants), posséder plusieurs types d'architecture logicielle.

L'architecture en flot de données est une série de fonctions dans un logiciel informatique où chaque étape est générée de manière automatique par les actions de la fonction précédente.

Dans ce cours, nous allons voir l'un de ces modèles d'architecture logicielle, l'architecture en flot de données, plus connue sous son nom anglais : *dataflow architecture*. Nous allons donc apprendre quand est-ce qu'on l'utilise, comment on l'utilise, pourquoi, ses avantages, etc. À la fin, vous posséderez donc toutes les clefs nécessaires pour pouvoir l'utiliser au mieux.

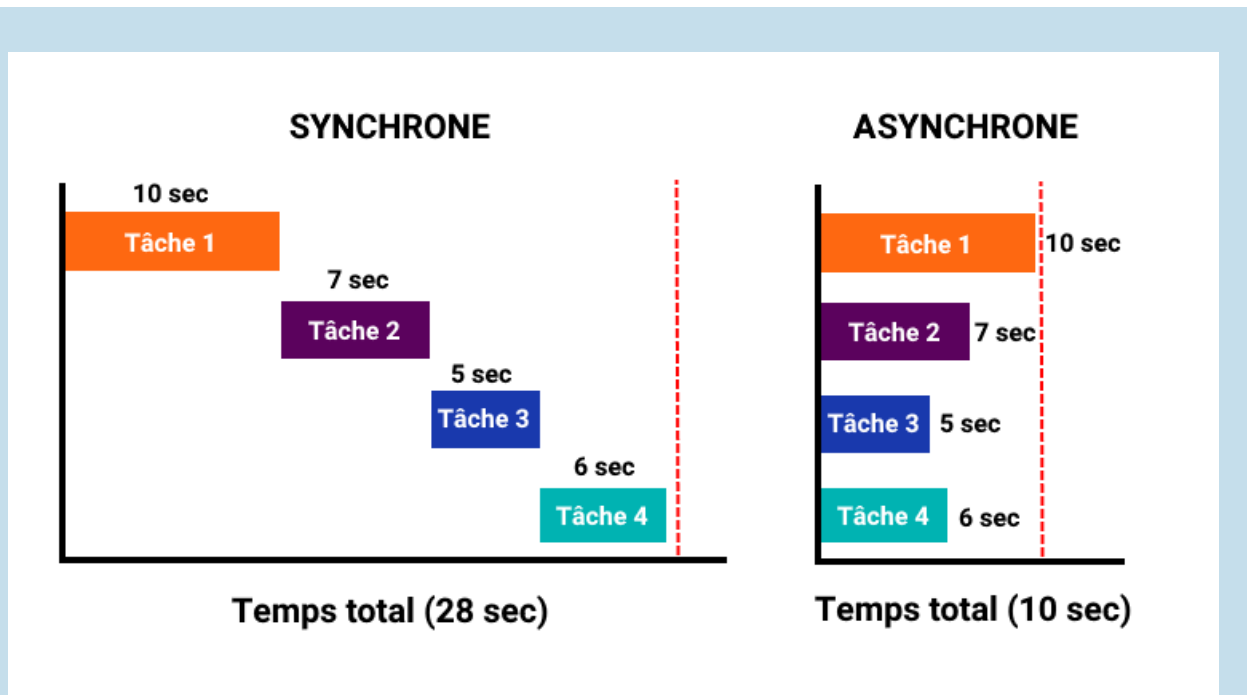
Définition

L'architecture en flot de données (en anglais : *dataflow architecture*) est une architecture où les données sont des entités actives traversant un programme de manière asynchrone. Dans un *dataflow*, les programmes sont représentés sous forme de graphes, l'opération à effectuer est représentée par un nœud, et les données formant les entrées aux nœuds circulent sur les arcs. Les données sont transportées par des jetons (aussi appelés *tokens*). La règle primordiale, dite « *de déclenchement* », se met en place uniquement lorsqu'un nœud voit toutes ses entrées satisfaites. Dès lors, il s'active et produit une valeur en sortie, et les jetons présents en entrée sont supprimés.

Une architecture de flux de données décrit le flux de travail suivi pour créer un système logiciel. Un flux de travail consiste en une série de transformations sur les informations d'entrée, où les informations et les méthodes de transformation sont indépendantes les unes des autres. Dans une architecture de flux de données, les informations sont extraites dans le système, puis transitent par plusieurs modules et sont transformées jusqu'à ce qu'elles atteignent leur destination (sortie ou stockage de données). Dans une architecture de flux de données, les transformations peuvent être réutilisées et modifiées.

Remarque

Quelque chose est dit asynchrone lorsque plusieurs tâches sont effectuées en même temps et à un rythme différent. Ce mode de fonctionnement fait face au mode synchrone qui, lui, doit attendre qu'une tâche soit finie pour en commencer une autre.



Légende : Exécution synchrone vs exécution asynchrone

Le transfert synchrone est efficace et fiable pour transférer de grandes quantités de données. Il fournit une communication en temps réel entre les appareils connectés, tels que le chat, la visioconférence, les conversations téléphoniques et les interactions en face à face. Synchrone fait référence au fait que le transfert est synchronisé avec une horloge.

Au contraire, les communications asynchrones sont non synchronisées avec une horloge, elles sont simples, bon marché et utilisées pour transférer de petites quantités de données. On privilégie ce mode de fonctionnement lorsque l'on veut communiquer avec quelqu'un sans que cela soit urgent, comme par exemple un message sur Slack.

Exemple

Un dataflow peut prendre plusieurs formes selon sa complexité, en revanche, il obéira toujours aux mêmes règles : l'architecture fonctionne comme une frise, elle se lit de gauche à droite de façon linéaire, et elle représente le chemin des transferts de données entre les différents composants d'une application.

L'architecture en flot de données est une architecture informatique qui se différencie directement avec l'architecture traditionnelle de Von Neumann. Dans une architecture traditionnelle, les flots de données attendent passivement dans l'espace mémoire pendant que le programme est exécuté de manière séquentielle. Ce type d'architecture matérielle était l'un des principaux axes de la recherche sur l'architecture informatique entre les années 1970 et les années 1980.

Définition

L'architecture de Von Neumann est un modèle pour une machine utilisant une structure de stockage unique, et ce dans le but de conserver à la fois les données demandées et les instructions ou bien alors produites suite à un calcul. Cette architecture porte ce nom en référence à un mathématicien américano-hongrois du nom de John

Von Neumann. Il a élaboré ce modèle d'architecture en juin 1945, qui donna par la suite la première description d'un ordinateur dont un programme est stocké dans sa mémoire. L'architecture de Von Neumann peut se décomposer en 4 points :

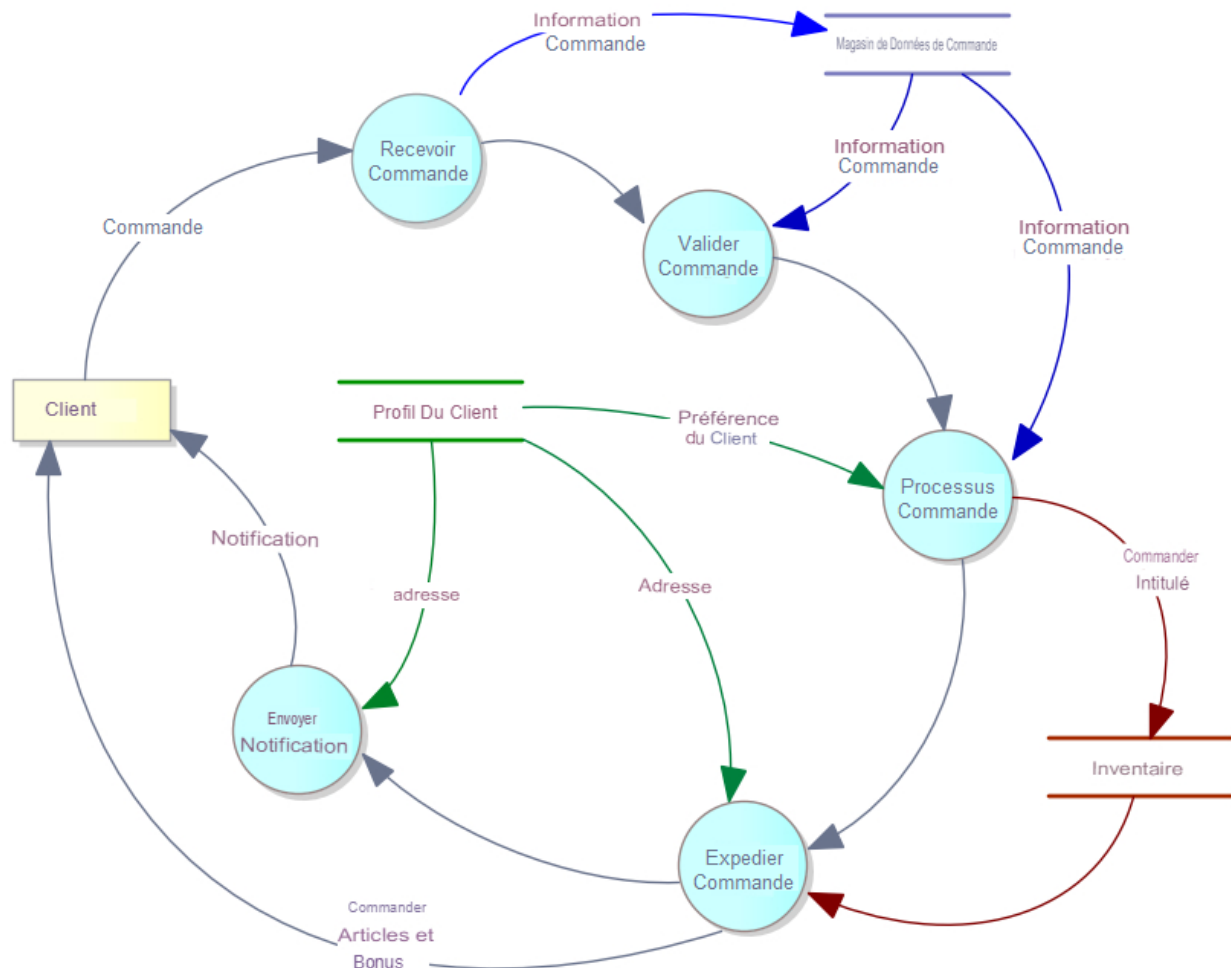
- Une mémoire contenant des données et des instructions,
- Une unité de contrôle en charge du séquençage (choisir l'ordre d'exécution) des opérations,
- Une unité arithmétique et logique afin d'effectuer les opérations basiques,
- Un jeu d'entrée / sortie pour la communication,

Les organigrammes sont utiles lorsque nous voulons décomposer un mécanisme ou un processus complexe en étapes simples, chaque étape représentant une étape d'un processus. Les étapes sont reliées de façon linéaire par des flèches du début à la fin du processus. Le but de la création d'un organigramme est de faciliter une meilleure compréhension par les employés, les examinateurs et le public.

Il fournit des informations qui permettent aux réviseurs de détecter et de résoudre les problèmes qui surviennent dans divers processus. D'autre part, les architectes de systèmes informatiques utilisent principalement des architectures en flot de données pour représenter le flux de données au sein du système. Cette forme de représentation graphique suit généralement une hiérarchie et peut comprendre plusieurs couches, selon la portée et la complexité du système représenté.

Remarque

Il y a une erreur à ne surtout pas faire, celle de confondre les architectures en flot de données (*Architecture dataflow*) et les diagrammes en flot de données (*Dataflow diagram* ou DFD). L'architecture en flot de données, comme son nom l'indique, sert à montrer l'architecture, le plus souvent matérielle et logicielle d'un projet, le diagramme en flot de données quant à lui, sert plutôt pour faire des schémas d'ordre plus général :



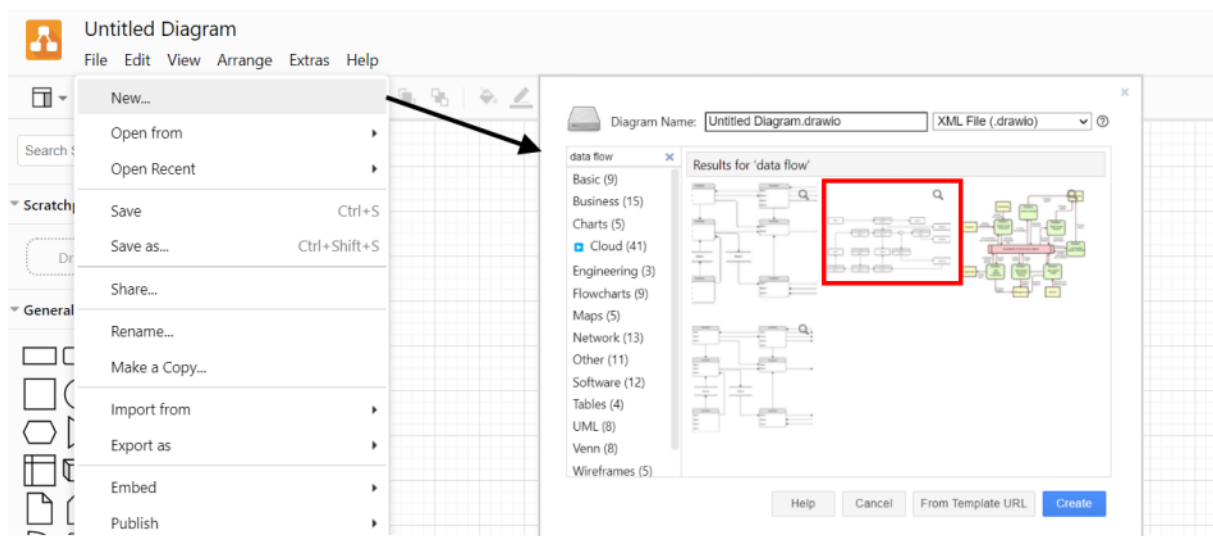
Légende : exemple d'un diagramme en flot de données

Définition

Comme nous pouvons le voir, il est assez différent d'une architecture en flot de données, notamment car il est beaucoup moins détaillé. Dans l'architecture en flot de données, l'ensemble du système logiciel a été pensé comme un ensemble de données d'entrée ou une série de transformations sur des éléments qui sont consécutifs, où les données et les opérations sont indépendantes les unes des autres. Dans cette approche, les données entrent dans le système puis à travers le module, une par une jusqu'à ce qu'elles soient affectées à une destination finale (sortie ou stockage de données).

Pour réaliser des architectures en flot de données il existe plusieurs sites internet, le problème étant qu'ils sont souvent payants. Pour ce faire, nous allons utiliser un outil que vous connaissez sûrement déjà : draw.io. L'avantage de ce site c'est déjà que nous n'avez pas besoin de posséder un compte pour faire vos diagrammes, il est très complet et totalement gratuit. De plus vous pouvez exporter vos créations sous plusieurs formats image ou document. En revanche, je vous conseille de partir d'un modèle déjà préétabli, puis de le modifier à votre guise, en effet, ce n'est pas le site le plus optimisé mais il satisfera quand même nos besoins. Voyons donc comment ouvrir ce modèle pour les diagrammes en flot de données.

Lorsque vous arrivez sur le site de draw.io vous allez choisir si vous voulez vous connecter ou non. Vous pouvez donc soit le lier avec votre compte et profiter d'une sauvegarde automatique en plus, soit cliquer en bas sur « *Decide later* » et commencer votre architecture. Comme dit précédemment, il vaut mieux partir d'un modèle déjà fait, or, lorsqu'on arrive sur le plan de travail de draw.io, on voit seulement un quadrillage avec quelques outils à droite. Pour ouvrir ce modèle, vous devez aller dans « *File* » → « *New...* » et rechercher « *dataflow* » :



Légende : choix du modèle « Dataflow 2 »

Source : Untitled Diagram¹

Une fois le modèle chargé, vous pouvez le modifier, ajouter des couleurs, faire des délimitations, etc. Mais attention, il est très important, encore plus que les couleurs et le fait que votre architecture soit jolie, il faut surtout qu'elle soit le plus lisible possible. Ne vous perdez donc pas à vouloir trop rajouter des choses futiles qui entacherait sa lisibilité.

Exercice : Quiz

[solution n°1 p.15]

Question 1

Par quoi peut-on traduire dataflow ?

- ☐ Flux de données
- ☐ Flot de données
- ☐ Courant de données

Question 2

¹ <https://app.diagrams.net/>

Les flots de données peuvent être des :

- ☐ Diagrammes
- ☐ Architectures
- ☐ Bases de données

Question 3

Une architecture en flot de données est différente d'un diagramme en flot de données.

- ☐ Vrai
- ☐ Faux

Question 4

Par quoi les données sont-elles transportées dans une architecture dataflow ?

- ☐ Des liens
- ☐ Des nœuds
- ☐ Des jetons

Question 5

Par quoi les opérations à effectuer sont-elles représentées ?

- ☐ Des liens
- ☐ Des nœuds
- ☐ Des jetons

III. Comment appliquer une architecture de données ?

Maintenant que nous avons vu ce qu'était une architecture en flot de données, voyons maintenant comment le faire. Tout d'abord il y a 2 types principaux d'architecture en flot de données :

- *Batch Sequential* (Lot séquentiel en français)
- *Pipe and Filter* (Tuyau et filtre en français)

Décortiquons donc ces deux architectures pour voir leurs différences, savoir quand et comment les utiliser. Nous allons commencer par le batch sequential, que nous appellerons le lot séquentiel pour plus de simplicité. Comme son nom l'indique, la tâche est divisée en plusieurs lots avec sa sous-tâche. Ces lots exécutent leurs sous-tâches et fournissent les résultats au lot suivant. C'est un modèle de traitement de données plutôt classique.

Nous pouvons utiliser le lot séquentiel, par exemple dans une transaction bancaire, où une fois que tous les détails requis pour la transaction sont fournis, seule la page suivante est traitée et le paiement peut alors être effectué.

Exemple

Le lot séquentiel peut prendre plusieurs formes selon la complexité de l'architecture choisie, il suivra néanmoins le même schéma :



Légende : Architecture en flot de données en lot séquentiel

Comme nous pouvons le voir, le lot séquentiel est assez facile à faire et marche de manière synchrone, nous allons donc voir ses avantages et ses inconvénients.

Définition

Comme dit précédemment, il faut voir les avantages et les inconvénients de ce modèle d'architecture en flot de données pour voir si celui-ci correspond à vos besoins et vos attentes. Voilà donc une liste :

Avantages :

- Les sous-systèmes ont des divisions simples,
- Les sous-systèmes sont indépendants des autres lots,
- Chaque lot transforme les données d'entrée et produit une sortie indépendante du lot adjacent,

Inconvénients :

- Il a une latence élevée car il est synchrone.
- Il y a une possibilité d'une surcharge de transformation de données entre les filtres.
- Sa maintenance n'est pas simple car demande un contrôle extérieur.

Maintenant que nous avons la première déclinaison de l'architecture en flot de données qui est le flot séquentiel, voyons maintenant l'autre forme : le Pipe and Filter. Cette approche s'appuie fortement sur la transformation de manière incrémentale des données via des composants séquentiels. Le flot de données est dirigé par les données ainsi que par l'ensemble du système qui est décomposé en composants de canaux, filtres, sources et récepteurs de données.

La connexion entre les modules est un flux de données, c'est un tampon entre le premier entré et le premier sorti, il peut s'agir d'un flux d'octets, d'un flux de caractères ou de tout autre type. La principale caractéristique de cette architecture est son exécution simultanée et incrémentale.

Définition

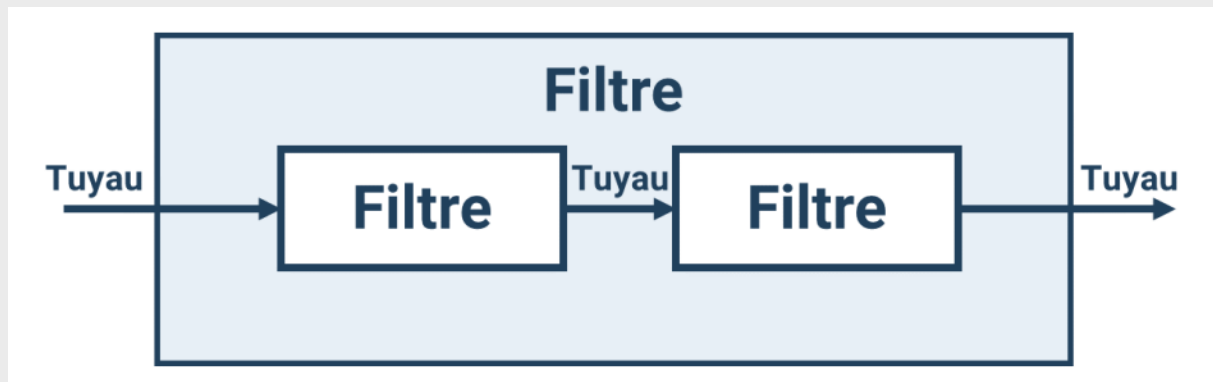
Par définition, une transformation de manière incrémentale des données via des composants séquentiels est une transformation par palier, petit à petit, des données, et cela grâce à des composants qui sont séquentiels. Un composant séquentiel est un composant dont les valeurs des variables de sortie dépendent des variables

d'entrée et de variables internes. Donc, pour résumer, une transformation de manière incrémentale des données via des composants séquentiels est une transformation par palier grâce à des composants dont leurs valeurs de sortie sont calculées par rapport aux valeurs d'entrée.

Maintenant que nous comprenons un peu mieux ce qu'une architecture en flot de données Pipe and Filter, quand l'utiliser ? Le Pipe and Filter est le meilleur pour les grands processus qui peuvent être décomposés en plusieurs étapes. De cette façon, chaque filtre sera responsable d'une des étapes et ils pourront s'exécuter simultanément pour produire toutes les données nécessaires au résultat final. Un des exemples courants d'application est le pipe sous Unix.

Exemple

Le Pipe est une commande sous Linux qui vous permet d'utiliser deux ou plusieurs commandes de sorte que la sortie d'une commande serve d'entrée à la suivante. En bref, la sortie de chaque processus va directement en entrée du suivant comme un pipeline. Sous UNIX / Linux, les filtres sont l'ensemble des commandes qui prennent l'entrée du flux d'entrée standard, effectuent certaines opérations et écrivent la sortie dans le flux de sortie standard. Voici un exemple de son architecture :



Légende : Architecture en flot de données Tuyau et Filtre

Un seul filtre a la possibilité de produire des données vers un ou plusieurs ports. Les filtres peuvent également s'exécuter simultanément et ne sont pas dépendants. La sortie d'un filtre est l'entrée d'un autre, par conséquent, l'ordre est très important.

Si vous pensiez tout savoir, il reste une chose à savoir avant de parler de ses avantages et de ses inconvénients, il existe deux types de filtres :

- Filtre actif
- Filtre passif

Le filtre actif sert à connecter les tuyaux pour rechercher des données et retirer les données transformées. Ce mode est utilisé dans le mécanisme que nous avons vu juste avant avec UNIX. Les filtres passifs, eux, permettent aux canaux connectés d'insérer et d'extraire des données. Il fonctionne avec un pipeline actif qui extrait les données d'un filtre et les pousse dans le filtre suivant. Il doit fournir un mécanisme de lecture / écriture.

Rappel

En informatique, un pipeline fait référence à une file d'attente logique qui est remplie de toutes les instructions à traiter en parallèle. Il s'agit du processus de stockage et de mise en file d'attente des tâches et des instructions qui sont exécutées simultanément par le processeur de manière organisée.

Les tuyaux ne possèdent pas d'état et ils transportent des flux binaires ou de caractères qui existent entre deux filtres. Un tuyau peut déplacer un flux de données d'un filtre à un autre. Ils utilisent peu d'informations contextuelles et ne conservent aucune information d'état entre les instanciations. Pour rappel l'instanciation désigne une programmation informatique qui se base sur un exemple déjà créé. On peut aussi l'utiliser pour effectuer un clonage, l'objet instancié pourra subir des modifications par la suite. Et donc pour finir, voyons les points positifs et les points négatifs de l'architecture en flot de données en tuyaux et filtres :

Avantages :

- Facilement modifiable.
- Fournit un accès simultané et un débit élevé pour un gros traitement des données.
- Une simplicité grâce à des divisions claires entre deux filtres connectés par un tuyau.
- Prend en charge l'exécution séquentielle et parallèle offrant une grande flexibilité.

Inconvénients :

- Ne permet pas aux filtres d'interagir en coopération pour résoudre un problème.
- Ne fonctionne que pour les interactions statiques.
- Le schéma dépend fortement de l'ordre des filtres.
- Il peut être difficile de concevoir des filtres incrémentiels.

Exercice : Quiz

[solution n°2 p.16]

Question 1

Combien existe-t-il de modèles d'architecture en flot de données ?

- ☐ 1
- ☐ 2
- ☐ 4

Question 2

Quel est le nom anglais du lot séquentiel ?

- ☐ Pack Sequential
- ☐ Sequential Lot
- ☐ Batch Sequential

Question 3

Le lot séquentiel fonctionne de manière asynchrone.

- ☐ Vrai
- ☐ Faux

Question 4

Quels inconvénients du lot séquentiel existent ?

- ☐ Sa maintenance demande un contrôle extérieur
- ☐ Il est asynchrone
- ☐ Il y a une possibilité d'une surcharge de transformation de données entre les filtres

Question 5

Le Pipe and Filter s'utilise pour des interactions statiques.

- ☐ Vrai
- ☐ Faux

V. Essentiel

L'architecture de flux de données décrit le flux de travail suivi pour créer un système logiciel. Le flux de travail consiste en une série de transformations sur les informations d'entrée, où les informations et les méthodes de transformation sont indépendantes les unes des autres. Dans une architecture de flux de données, les informations sont extraites dans le système qui circule ensuite à travers plusieurs modules et subit une transformation jusqu'à ce que la destination (sortie ou magasin de données) soit atteinte. Dans l'architecture de flux de données, les transformations peuvent être réutilisées et modifiées.

Les connexions entre composants ou modules peuvent être implémentées sous forme de flux d'E / S (entrée / sortie), de tampons d'E / S, de canaux ou d'autres types de connexions. Les données peuvent être transférées dans des topologies du graphique avec cycles, dans des structures linéaires ou arborescentes sans cycles.

L'objectif principal de cette approche est d'atteindre une qualité de réutilisabilité et de modifiabilité. Il convient aux applications impliquant une série de transformations de données indépendantes bien définies ou de calculs sur des entrées et des sorties définies et surtout ordonnées, comme les compilateurs et les applications de traitement de données d'entreprise.

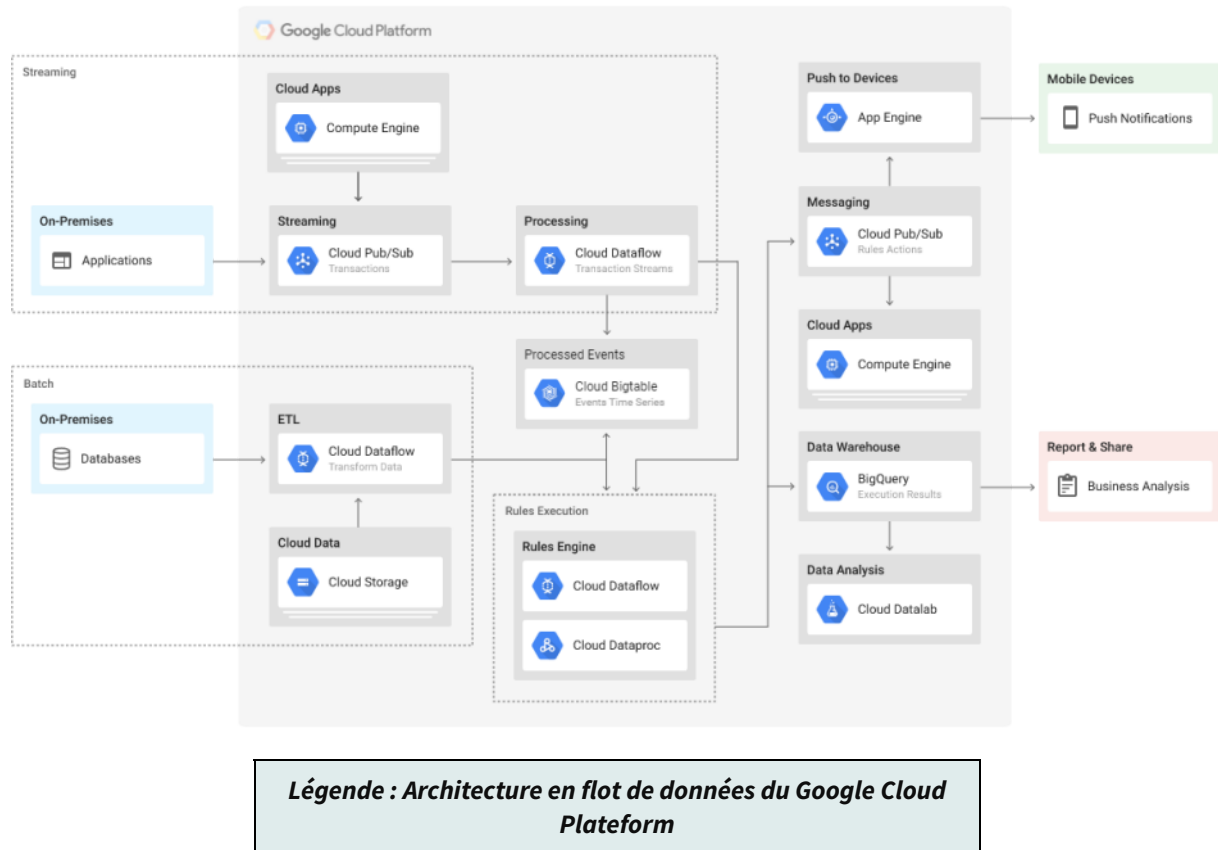
Étant donné que l'architecture de flux de données est considérée comme une forme de programmation assez simple, elle est souvent utilisée par des programmeurs débutants comme expérimentés. En utilisant cette méthode, le système peut être programmé paquet par paquet. Bien que l'architecture de flot de données soit relativement facile à utiliser, il existe certaines limitations, avec des composants dédiés à des tâches spécifiques, le système peut fonctionner efficacement, mais avec seulement une certaine flexibilité. Chaque composant peut être programmé différemment, mais pour que le système fonctionne, il doit suivre un certain chemin, cela limite donc un petit peu votre champ de liberté.

Une fois que vous maîtrisez à la perfection la création de différentes architectures en flot de données, qu'elles soient en Batch Sequential (lots séquentiels) ou bien en Pipes and Filters (tuyaux et filtres), cet outil va se révéler comme un atout indispensable et incontournable. Il est facilement lisible et compréhensible par tout le monde, comme vos collaborateurs, qu'ils soient informaticiens ou complètement novice, si votre architecture en flot de données est bien réalisée. Cela entraînera donc un gain de temps énorme car tout le monde comprendra rapidement les objectifs et les missions d'un projet.

VI. Auto-évaluation

A. Exercice

Google est une entreprise qui donne beaucoup de ses données comme des API, des applications, etc. en open source, c'est-à-dire gratuites et visibles par tous. Par chance, Google a rendu publiques leurs architectures en flot de données de leur service Cloud qui s'appelle le Google Cloud Platform. Soit l'image ci-dessous :

Source¹

Question 1

[solution n°3 p.17]

Avec cette image, déterminer s'il s'agit d'un modèle en lot séquentiel ou bien d'un modèle en tuyau et filtre. Expliquez pourquoi.

Question 2

[solution n°4 p.17]

Reproduisez cette architecture sur draw.io.

B. Test

Exercise 1 : Quiz

[solution n°5 p.17]

Question 1

Dans les lots séquentiels, quel filtre existe ?

- ☐ Filtre de calcul
- ☐ Filtre possessif
- ☐ Filtre passif

1 <https://cloud.google.com/architecture/images/architecture-complex-event-processing.svg>

Question 2

Les architectures en flot de données et les diagrammes en flot de données sont la même chose.

- ☐ Vrai
- ☐ Faux

Question 3

Quels sont les avantages du lot séquentiel ?

- ☐ Il est modifiable assez facilement
- ☐ Il fonctionne que pour les interactions statiques et dynamiques
- ☐ Il est plus facilement lisible

Question 4

La commande pipe d'UNIX vient de la même origine que le Pipe and Filter.

- ☐ Vrai
- ☐ Faux

Question 5


Quelles caractéristiques possèdent les tuyaux ?

- ☐ Transportent des flux binaires
- ☐ Ne possèdent pas d'état
- ☐ Possèdent un état logique

Solutions des exercices


Exercice p. 7 Solution n°1**Question 1**

Par quoi peut-on traduire dataflow ?

- ☒ Flux de données
- ☐ Flot de données
- ☐ Courant de données
-  Dataflow peut se traduire par flux de données.


Question 2

Les flots de données peuvent être des :

- ☒ Diagrammes
- ☒ Architectures
- ☐ Bases de données
-  Les flots de données peuvent être représentés par des diagrammes et des architectures.


Question 3

Une architecture en flot de données est différente d'un diagramme en flot de données.

- ☒ Vrai
- ☐ Faux
-  L'architecture en flot de données montre principalement les échanges de données à travers les logiciels et le matériel alors que le diagramme en flot de données reprend ce concept mais de manière beaucoup plus générale.

Question 4


Par quoi les données sont-elles transportées dans une architecture dataflow ?

- ☐ Des liens
- ☐ Des nœuds
- ☒ Des jetons
-  Dans une architecture en flot de données, les données sont transportées par des jetons, plus communément appelés tokens.

Question 5

Par quoi les opérations à effectuer sont-elles représentées ?

- ☐ Des liens
- ☒ Des nœuds
- ☐ Des jetons


 Dans une architecture en flot de données, les opérations à effectuer sont représentées par des nœuds.

Exercice p. 11 Solution n°2

Question 1

Combien existe-t-il de modèles d'architecture en flot de données ?


- ☐ 1
- ☒ 2
- ☐ 4

 Il en existe deux, le Pipe and Filter et le lot séquentiel.

Question 2

Quel est le nom anglais du lot séquentiel ?


- ☐ Pack Sequential
- ☐ Sequential Lot
- ☒ Batch Sequential

 La traduction du lot séquentiel est Batch Sequential.

Question 3

Le lot séquentiel fonctionne de manière asynchrone.


- ☐ Vrai
- ☒ Faux

 Le lot séquentiel marche de manière synchrone.

Question 4

Quels inconvénients du lot séquentiel existent ?

- ☒ Sa maintenance demande un contrôle extérieur
- ☐ Il est asynchrone
- ☒ Il y a une possibilité d'une surcharge de transformation de données entre les filtres

 Le lot séquentiel demande en effet un contrôle extérieur pour sa maintenance la rendant donc plus compliquée, mais il a aussi s'il est mal conçu une possibilité de surcharge. En revanche, il n'est pas asynchrone.

Question 5

Le Pipe and Filter s'utilise pour des interactions statiques.

☒ Vrai

☐ Faux

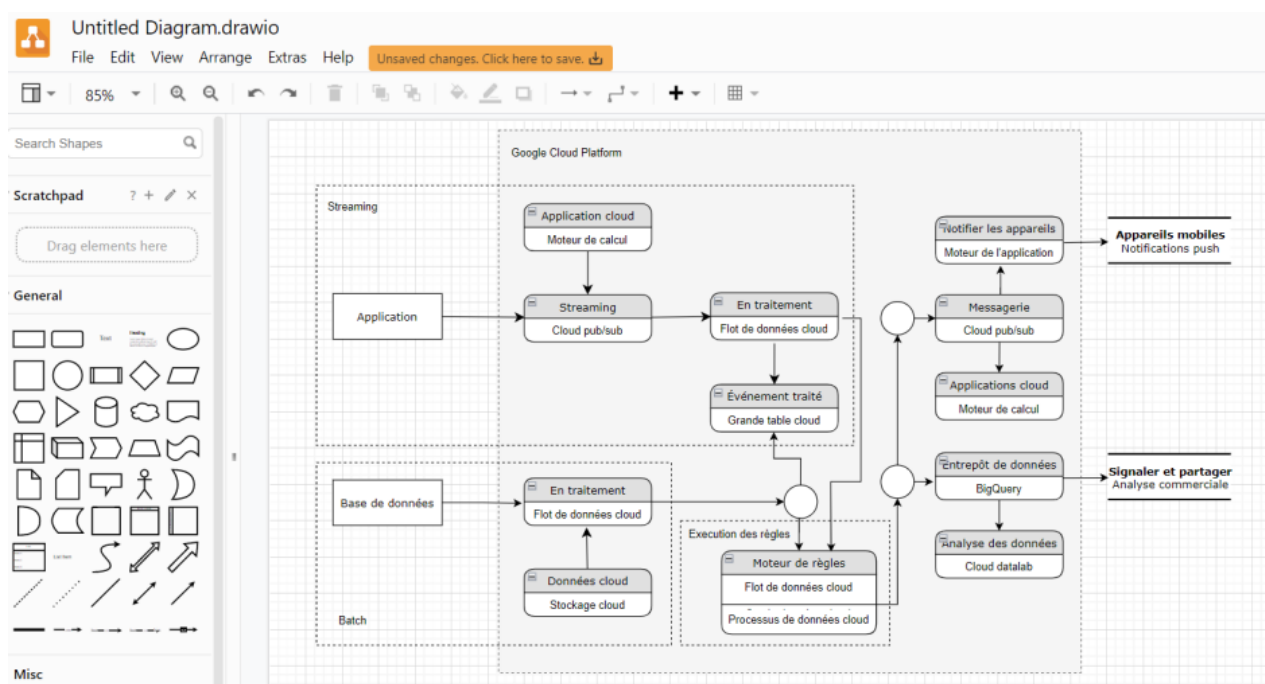
Ce modèle ne permet pas de prendre en charge les interactions de type dynamiques.

p. 13 Solution n°3

Cette architecture en flot de données utilise le modèle des tuyaux et des filtres, on peut le reconnaître grâce à trois indicateurs :

- La présence de composants (tous les rectangles comme Streaming, ETL, Processing, etc.)
- La présence de tuyaux (ici représenté par des flèches)
- La présence de filtres (ce sont les rectangles en pointillés comme Streaming et Batch)

p. 13 Solution n°4



Légende : Architecture en flot de données du Google Cloud Platform avec le modèle de tuyau et de filtre

Source : Untitled Diagram¹


Exercice p. 13 Solution n°5

¹ <https://app.diagrams.net/>

Question 1

Dans les lots séquentiels, quel filtre existe ?


- ☐ Filtre de calcul
- ☐ Filtre possessif
- ☒ Filtre passif

 Il existe deux types de filtres, les filtres actifs et passifs. Les filtres passifs sont là pour insérer et extraire les données des tuyaux qui sont connectés.

Question 2

Les architectures en flot de données et les diagrammes en flot de données sont la même chose.


- ☐ Vrai
- ☒ Faux

 Bien que similaires, les Diagrammes en Flot de Données (ou DFD) sont plus généraux et moins précis.

Question 3

Quels sont les avantages du lot séquentiel ?


- ☒ Il est modifiable assez facilement
- ☐ Il fonctionne que pour les interactions statiques et dynamiques
- ☒ Il est plus facilement lisible

 Du fait qu'il n'a pas besoin d'aide extérieure pour sa maintenance, ça le rend assez facilement modifiable. De plus, sa structure plutôt simple le rend plus lisible. En revanche, le lot séquentiel ne fonctionne pas avec les interactions dynamiques.

Question 4

La commande pipe d'UNIX vient de la même origine que le Pipe and Filter.


- ☒ Vrai
- ☐ Faux

 La commande pipe suit le raisonnement du modèle de tuyau et de filtre c'est-à-dire qu'elles effectuent certaines opérations et écrivent la sortie dans le flux de sortie standard.

Question 5

Quelles caractéristiques possèdent les tuyaux ?

- ☒ Transportent des flux binaires
- ☒ Ne possèdent pas d'état
- ☐ Possèdent un état logique

 Les tuyaux ne possèdent pas d'état et servent à transporter des flux binaires entre les composants.