

EGC 2007

Tutorial Gestion et Fouille de Flux de Données

23 Janvier 2007, Namur, Belgique

Intervenants

Fabrice Clérot (France Télécom R&D)

Baptiste Csernel (France Télécom R&D et Télécom Paris)

Georges Hébrail (Télécom Paris)



Plan

- Système d'information et flux de données
 - Fabrice CLEROT
- Systèmes de gestion des flux de données
 - Georges HEBRAIL
- Techniques d'analyse de flux de données
 - en un point d'observation, Baptiste CSERNE
 - distribuées, Fabrice CLEROT

Plan

- Système d'information et flux de données
 - Fabrice CLEROT
- Systèmes de gestion des flux de données
 - Georges HEBRAIL
- Techniques d'analyse de flux de données
 - en un point d'observation, Baptiste CSERNE
 - distribuées, Fabrice CLEROT

What we're up against

Randy SHOUP, Dan PRITCHETT,
"The eBay Architecture", SD Forum 2006

- eBay manages ...

- Over 212,000,000 registered users
- Over 1 Billion photos

- eBay users worldwide trade more than \$1590 worth of goods every second
- eBay averages over 1 billion page views per day
- At any given time, there are approximately 105 million listings on the site
- eBay stores over 2 Petabytes of data – over 200 times the size of the Library of Congress!
- The eBay platform handles 3 billion API calls per month

- In a dynamic environment

- 300+ features per quarter
- We roll 100,000+ lines of code every two weeks

- In 33 countries, in seven languages, 24x7

>26 Billion SQL executions/day!



Over ½ Million pounds of
Kimchi are sold every year!

Auction
www.auction.co.kr
an eBay company



© 2006 eBay Inc.

Massive Scale of Data

MassDAL
Massive Data Analysis Lab

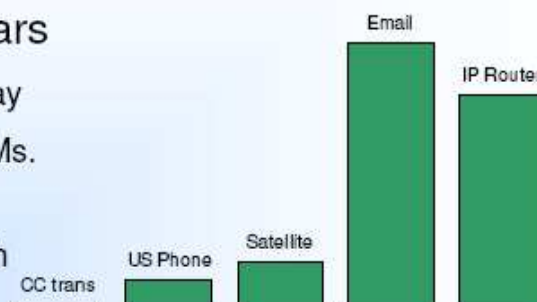
Explosion of Data In Recent Years

- 3 Billion Telephone Calls in US each day
- 30 Billion emails daily, 1 Billion SMS, IMs.

• **Scientific data:** NASA's observation satellites generate billions of readings each per day.

• **IP Network Traffic:** up to 1 Billion packets per hour per router. Each ISP has many (hundreds) of routers!

• **Compare to "human scale" data:** "only" 1 billion worldwide credit card transactions per month.



New data scales demand new approaches from databases, algorithms, networks, systems and engineering.

Muthu Muthukrishnan, ~~Rutgers Univ.~~
Google

ystème d'information et flux de données

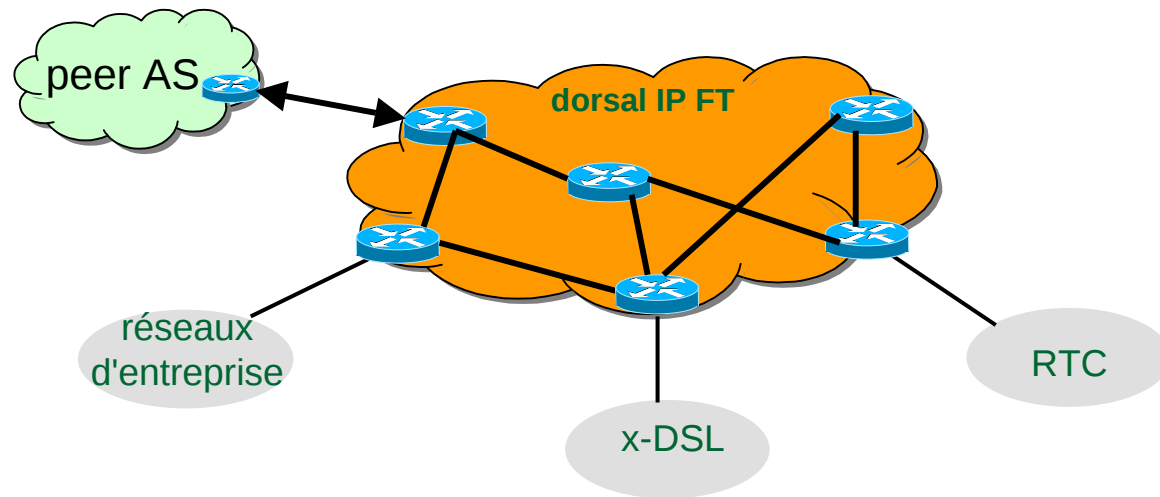
- deux exemples de flux de données
- définition d'un flux de données: observation et modélisation
- flux de données et système d'information
- contraintes caractéristiques du traitement des flux de données

système d'information et flux de données

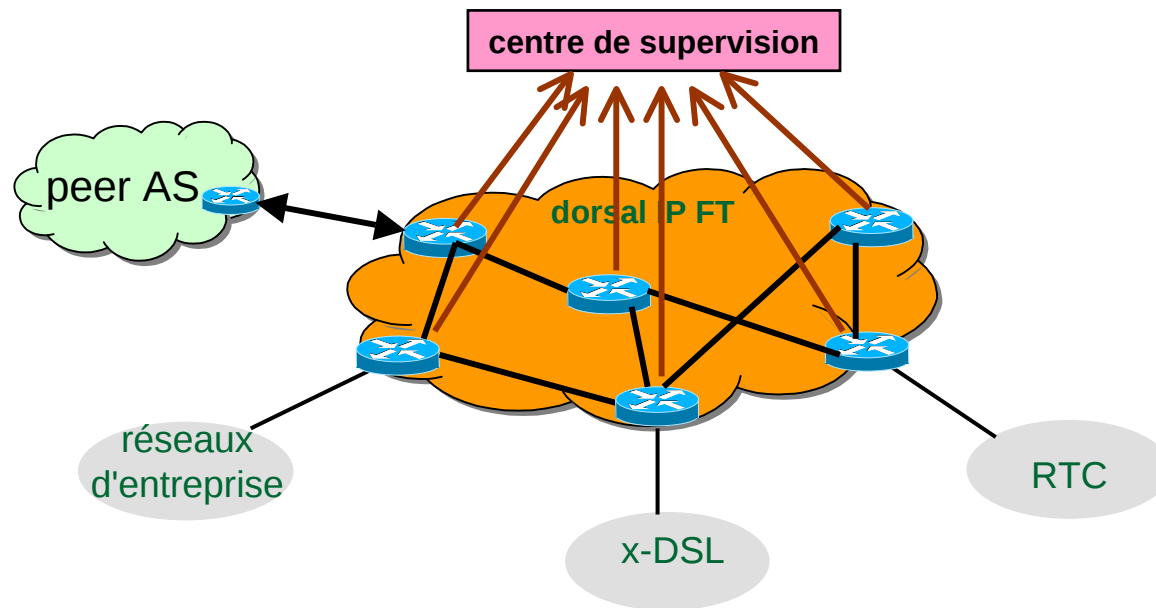
- deux exemples de flux de données
- définition d'un flux de données: observation et modélisation
- flux de données et système d'information
- contraintes caractéristiques du traitement des flux de données

exemple 1: supervision de réseau télécom

exemple 1: supervision de réseau télécom

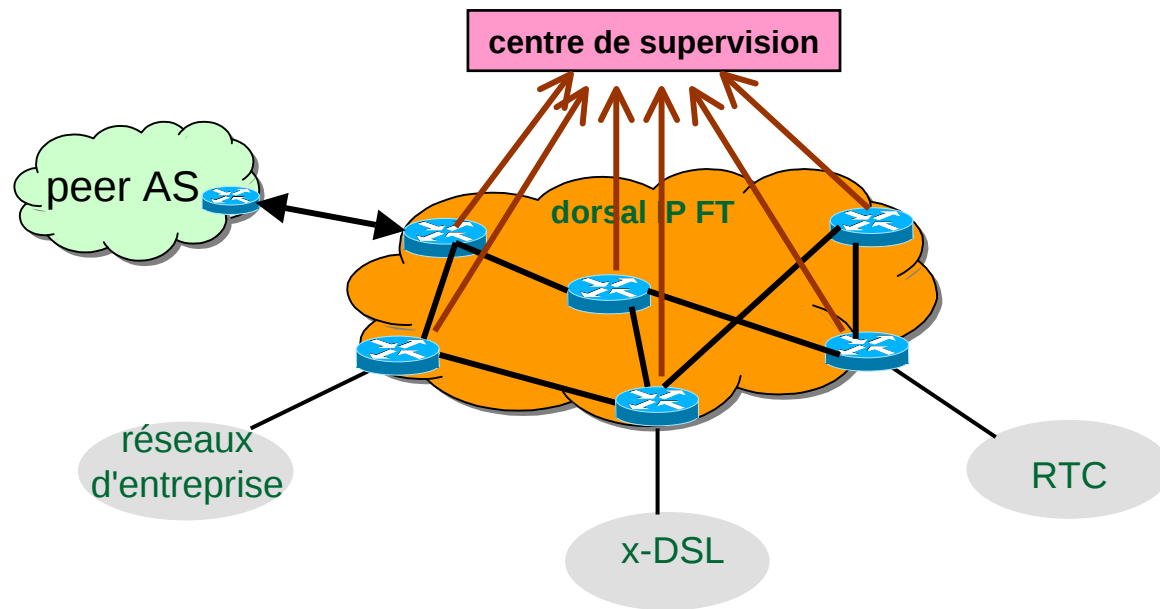


exemple 1: supervision de réseau télécom



remontée des enregistrements
NetFlow par les routeurs du
dorsal vers le centre de
supervision

exemple 1: supervision de réseau télécom

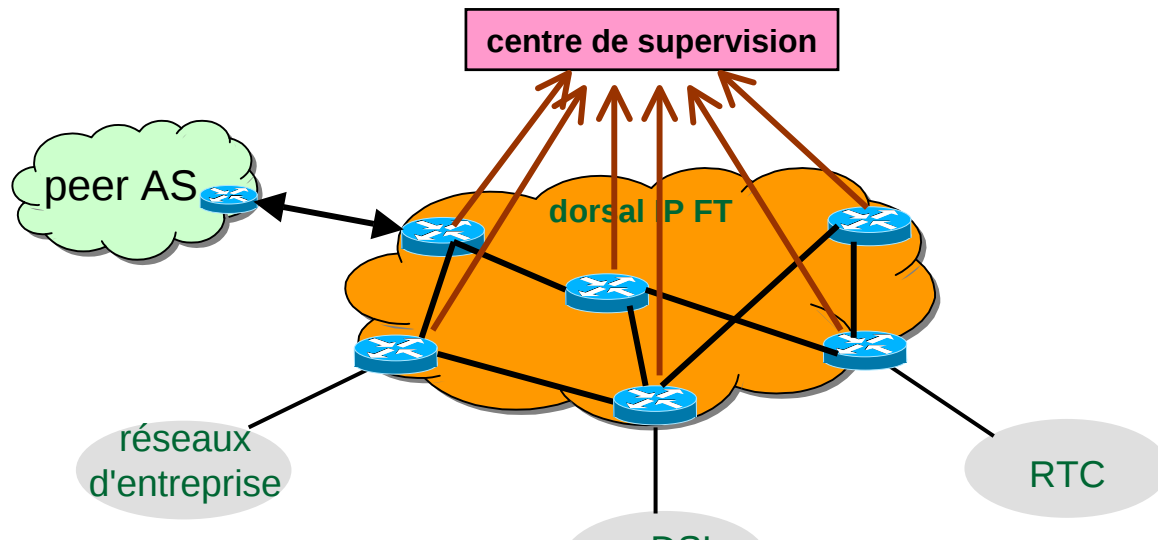


remontée des enregistrements
NetFlow par les routeurs du
dorsal vers le centre de
supervision

un enregistrement NetFlow = caractérisation d'un flot observé
@S, @D, pS, pD, Protocole, NbPaquets, NbOctets, Durée, ...

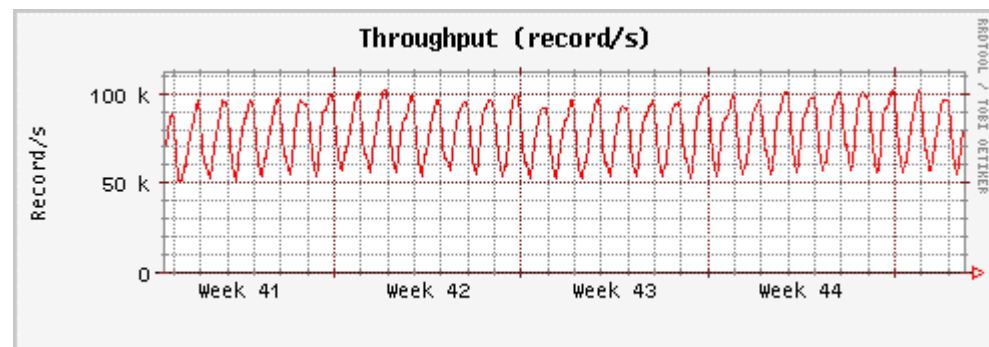
sonde NetFlow: sonde embarquée (propriétaire CISCO)

exemple 1: supervision de réseau télécom

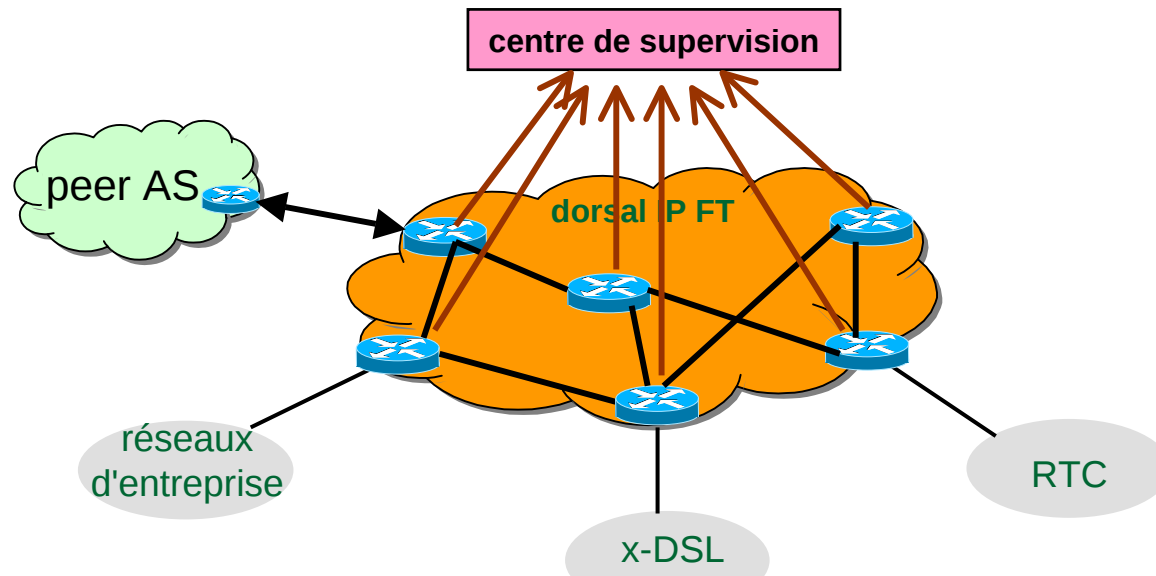


remontée des enregistrements
NetFlow par les routeurs du
dorsal vers le centre de
supervision

les volumes collectés sont
énormes (~ 300 Go / jour)
le débit des enregistrements
est très élevé (~ 75000 /s)
les sondes échantillonnent le
trafic (1 paquet sur 100) !



Exemple 1: Supervision de réseau télécom



remontée des enregistrements
NetFlow par les routeurs du
dorsal vers le centre de
supervision

les 1000 (@S, @D) les plus fréquentes sur R1...

combien de (@S, @D) différents ont été vues sur R1 et pas R2 ...

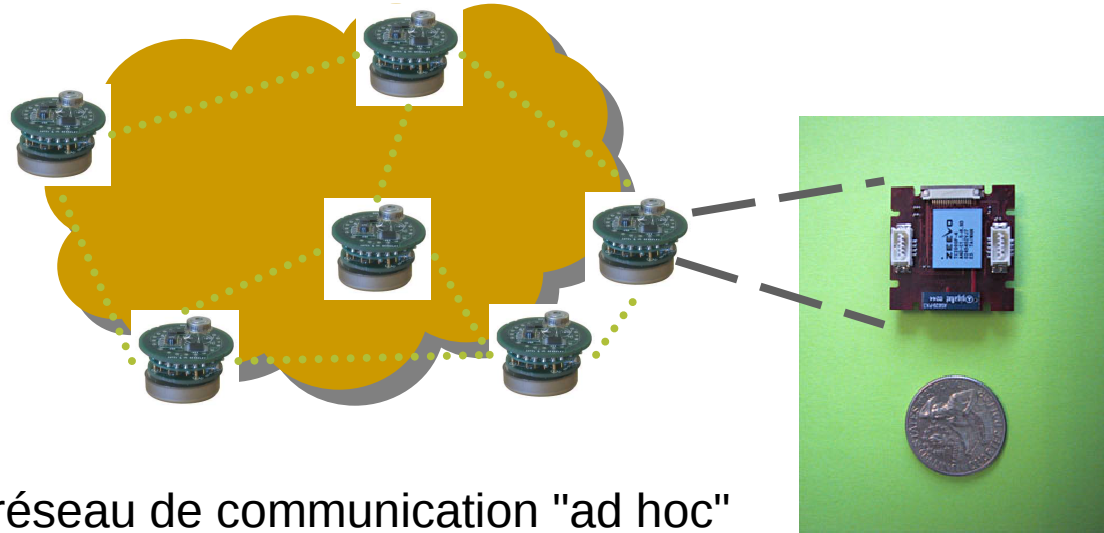
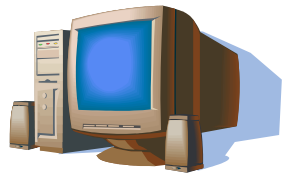
combien d' @ sont @S sur R1 et @D sur R2 ...

... sur le mois, sur la semaine, *sur la journée, sur l'heure ?*

... sur une période donnée du passé ?

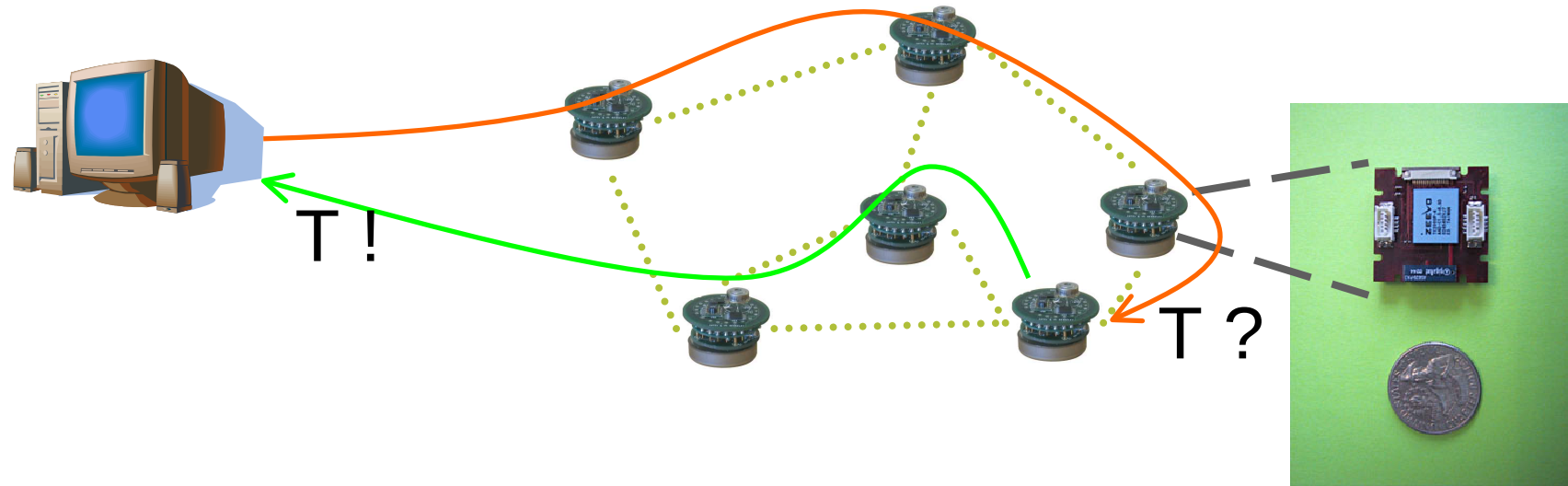
exemple 2: réseau de capteurs

exemple 2: réseau de capteurs

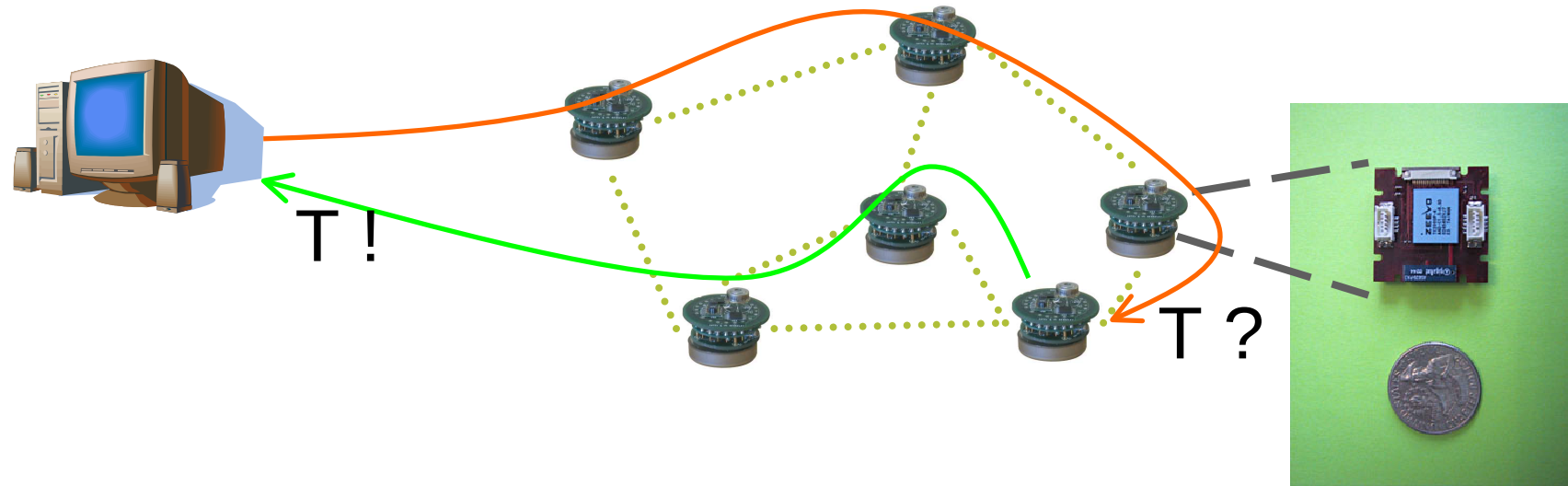


réseau de communication "ad hoc"

exemple 2: réseau de capteurs

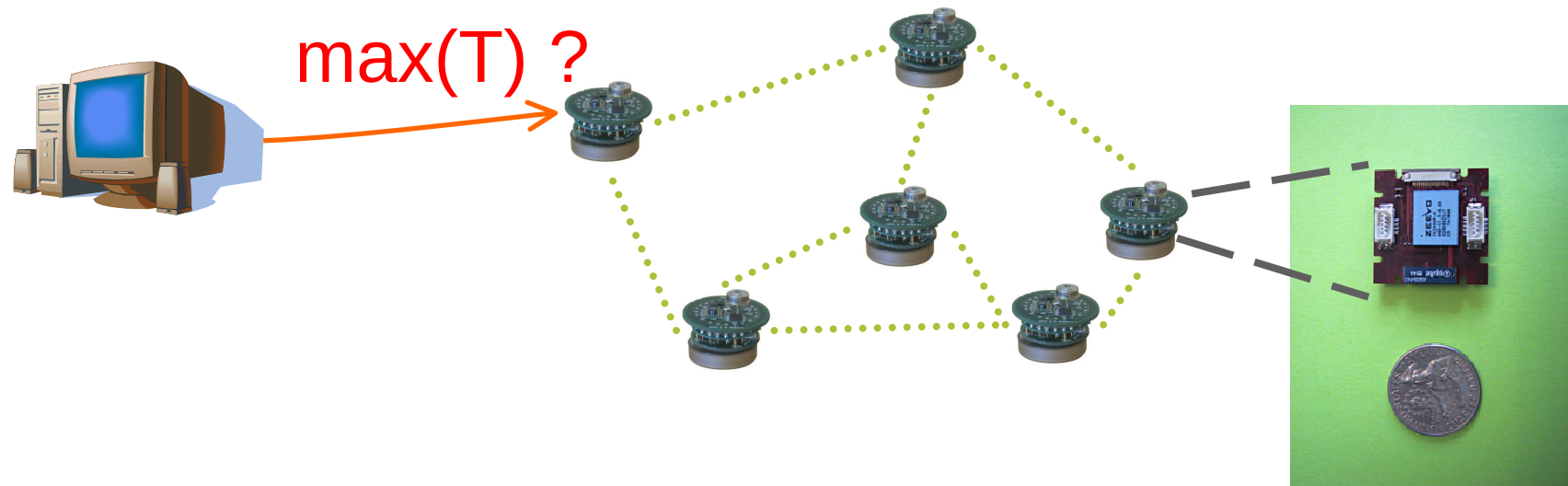


exemple 2: réseau de capteurs



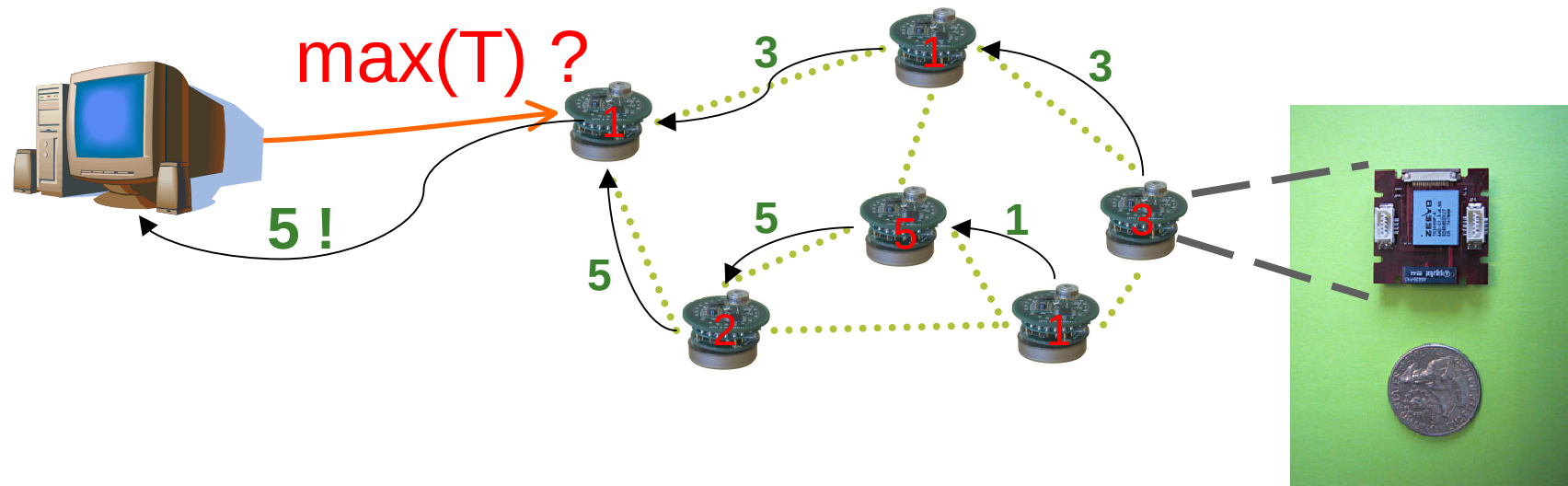
- les volumes sont faibles ...
- ... mais les capteurs sont (très) nombreux
- ... et les ressources sont très limitées
 - ❑ énergie
 - ❑ mémoire, puissance de calcul, bande passante

exemple 2: réseau de capteurs



- les volumes sont faibles ...
- ... mais les capteurs sont (très) nombreux
- ... et les ressources sont très limitées
 - énergie
 - mémoire, puissance de calcul, bande passante

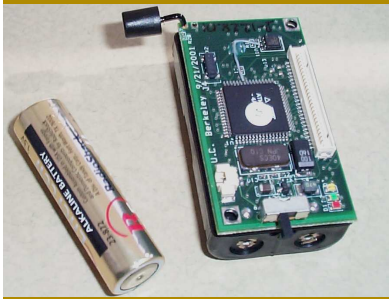
exemple 2: réseau de capteurs



- les volumes sont faibles ...
- ... mais les capteurs sont (très) nombreux
- ... et les ressources sont très limitées
- traiter les données dans le réseau ad hoc permet de réduire la consommation de ressources

"motes": plus que de simples capteurs

Crossbow
TECHNOLOGY INC.



Mica Mote



Mica2Dot

uProc: 4Mhz, 8 bit Atmel RISC

Radio:

40 kbit 900/450/300 MHz *or*
250 kbit 2.5GHz (MicaZ 802.15.4)

Memory:

4 K RAM / 128 K Program Flash /
512 K Data Flash

Power: 2 x AA or coin cell

Samuel Madden

<http://db.lcs.mit.edu/madden/>

Intel Research

iMote (1 + 2)



uProc: 12Mhz, 16 bit ARM

Radio: Bluetooth / 802.15.4

Memory:

64K SRAM / 512 K Data Flash

Power: 2 x AA

capteurs

- montés sur carte-fille
 - ❑ météo (température, éclairement, pression, humidité)
 - ❑ vibration (accéléromètres 2 ou 3 axes)
 - ❑ tracking (microphone, magnétomètre)
 - ❑ GPS
 - ❑ lecteur RFID



durée de vie

- batterie
 - 1000-2000 mA.h
- consommation variable en fonction de l'application
 - processeur Mica2:
 - 5 mA active, 1 mA idle, 5 μ A sleep
 - radio CC1000
 - 5 mA listen, 10 mA xmit/receive, ~20 ms / packet
 - capteur proprement dit
 - 1 μ A -> 100's mA, 1 μ s -> 1 s / sample

ystème d'information et flux de données

- deux exemples de flux de données
- définition d'un flux de données: observation et modélisation
- flux de données et système d'information
- contraintes caractéristiques du traitement des flux de données

définition d'un flux de données

- distinguer observation et modélisation
 - observation: données qu'on lit sur le flux
 - modélisation: signal qu'on cherche à reconstruire à partir de ces données

observation d'un flux de données

- suite infinie d'instances $x_i = (t_i, m_i)$
 - t_i l'instant d'observation
 - temps "physique"
 - temps "du flux": $t_i = i$
 - m_i un ensemble de données descriptives numériques ou symboliques (dont des identifiants)
- exemple:
 - observation à t_i d'un paquet :
 - $m_i = (@S, @D, pS, pD, Protocole, Volume)$

modèles de flux de données

- le modèle relie l'observation du flux à un signal sous-jacent
 - observation $m: \mathbf{N} \rightarrow A \times \mathbf{R}$
 - A = espace d'identifiants et $m(i)=(a, v)$
 - signal $M: A \times \mathbf{T} \rightarrow \mathbf{R}$
 - reconstruire une information sur l'ensemble des identifiants à partir de l'observation du flux
 - $\mathbf{T}$: temps, implicite ou explicite
 - $M(\bullet, t_i) \leftarrow F(M(\bullet, t_{i-1}), m_i)$
 - difficile si $|A|$ est très grand
- plusieurs modèles possibles pour un même flux

modèles "canoniques" de flux de données

- deux dimensions distinctes
 - un temps
 - un espace d'adresse
- série temporelle
- cumul par adresse
 - " tiroir-caisse " ("cash register")
 - cumul pour chaque identifiant de valeurs positives
 - "tourniquet" ("turnstile")
 - cumul pour chaque identifiant de valeurs positives ou négatives
- 3 modèles parmi une infinité ...

exemple

- Flux de données:
 - observation à l'instant i d'un paquet de volume v émis par la source a
 - $m_i = (a, v)$

exemple

■ modèle 1:

- ❑ série temporelle des volumes des paquets, toutes @S confondues:
- ❑ observation de $m_i = (a, v) \rightarrow M(i) = v$
- ❑ série temporelle

■ modèle 2:

- ❑ cumul des volumes des paquets par @S
- ❑ observation de $m_i = (a, v) \rightarrow M(a) = v$
- ❑ " tiroir caisse "

exemple

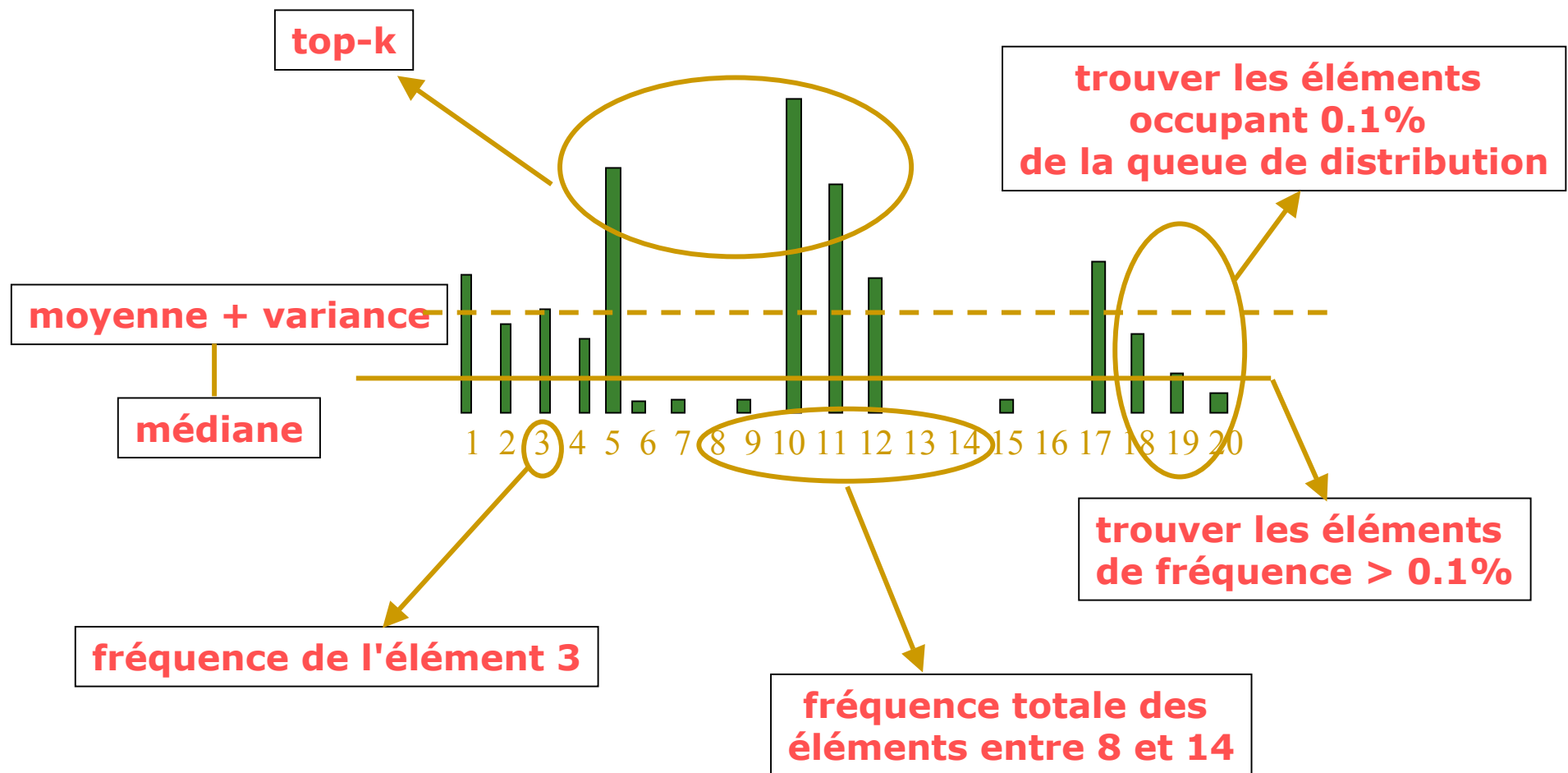
■ modèle 3:

- ❑ cumul des volumes des paquets, toutes @S confondues:
- ❑ observation de $m_i = (a, v) \rightarrow M = M + v$
- ❑ " tiroir caisse " (?)

■ modèle 4:

- ❑ séries temporelles des cumuls des volumes des paquets par @S
- ❑ observation de $m_i = (a, v) \rightarrow M(a, i) = M(a, i) + v$
- ❑ " tiroir-caisse " et série temporelle

exemples de requêtes (tiroir-caisse)



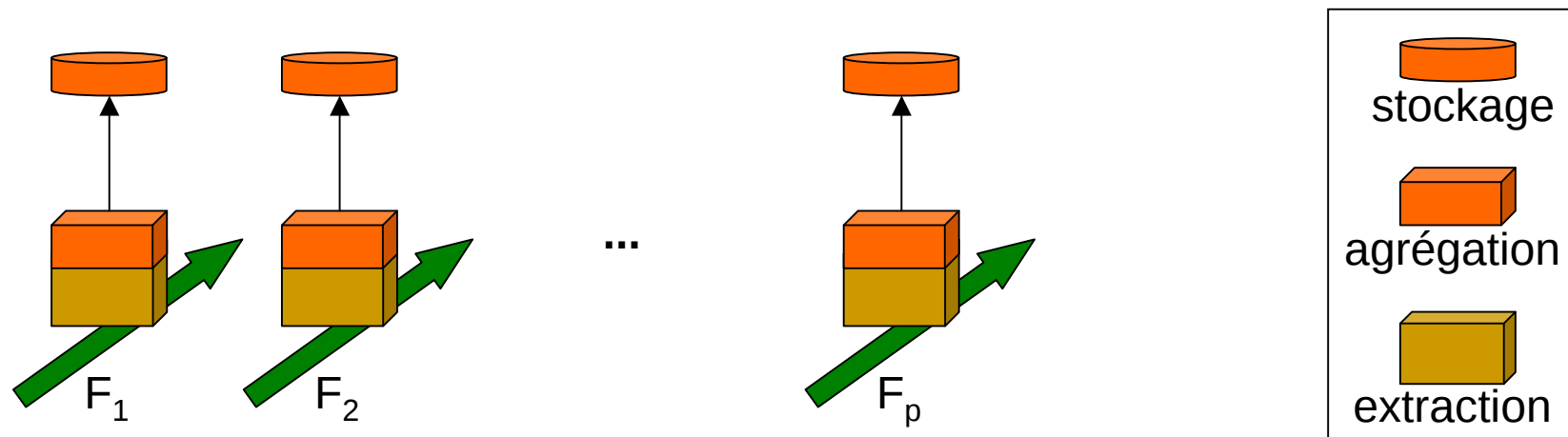
nombre d'éléments ayant une fréquence non nulle

système d'information et flux de données

- deux exemples de flux de données
- définition d'un flux de données: observation et modélisation
- flux de données et système d'information
- contraintes caractéristiques du traitement des flux de données

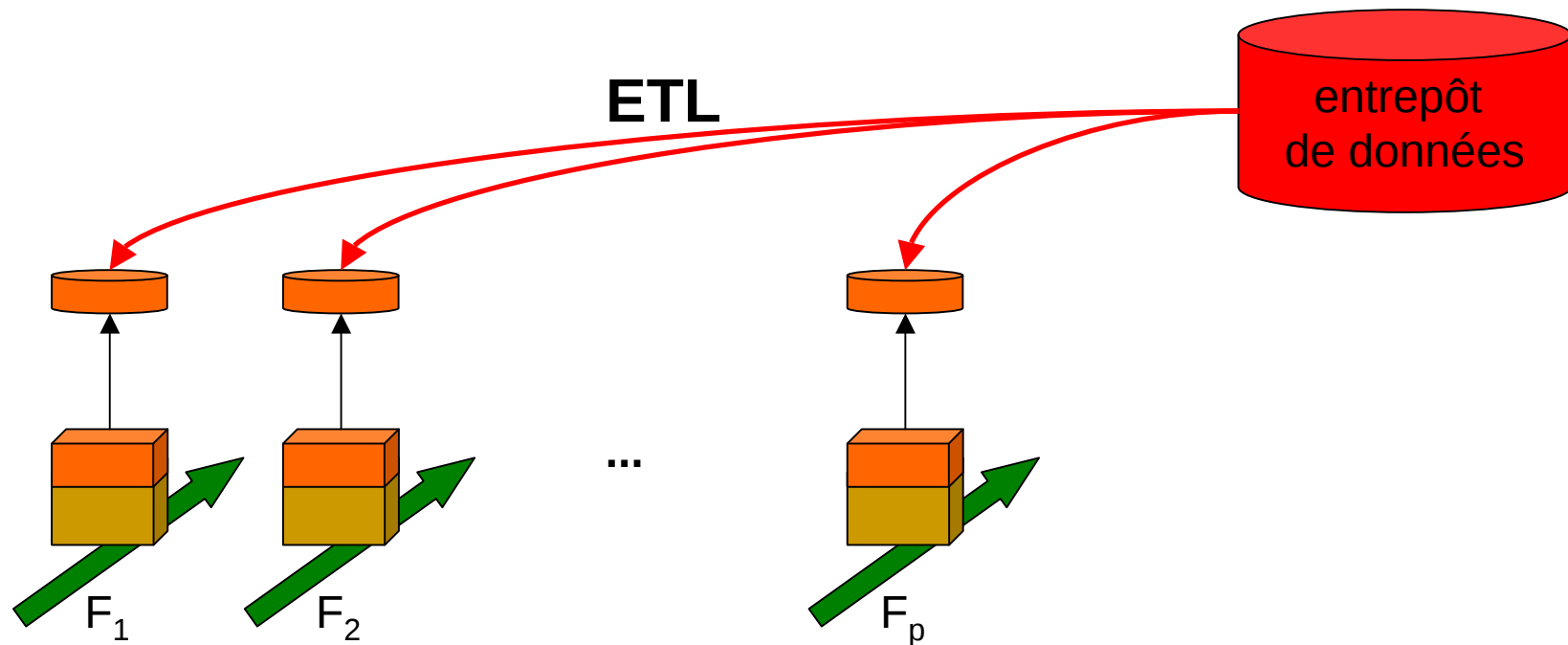
flux de données et système d'information

- constitution d'agrégats prédéfinis par des systèmes propriétaires
- stockage temporaire, local des agrégats



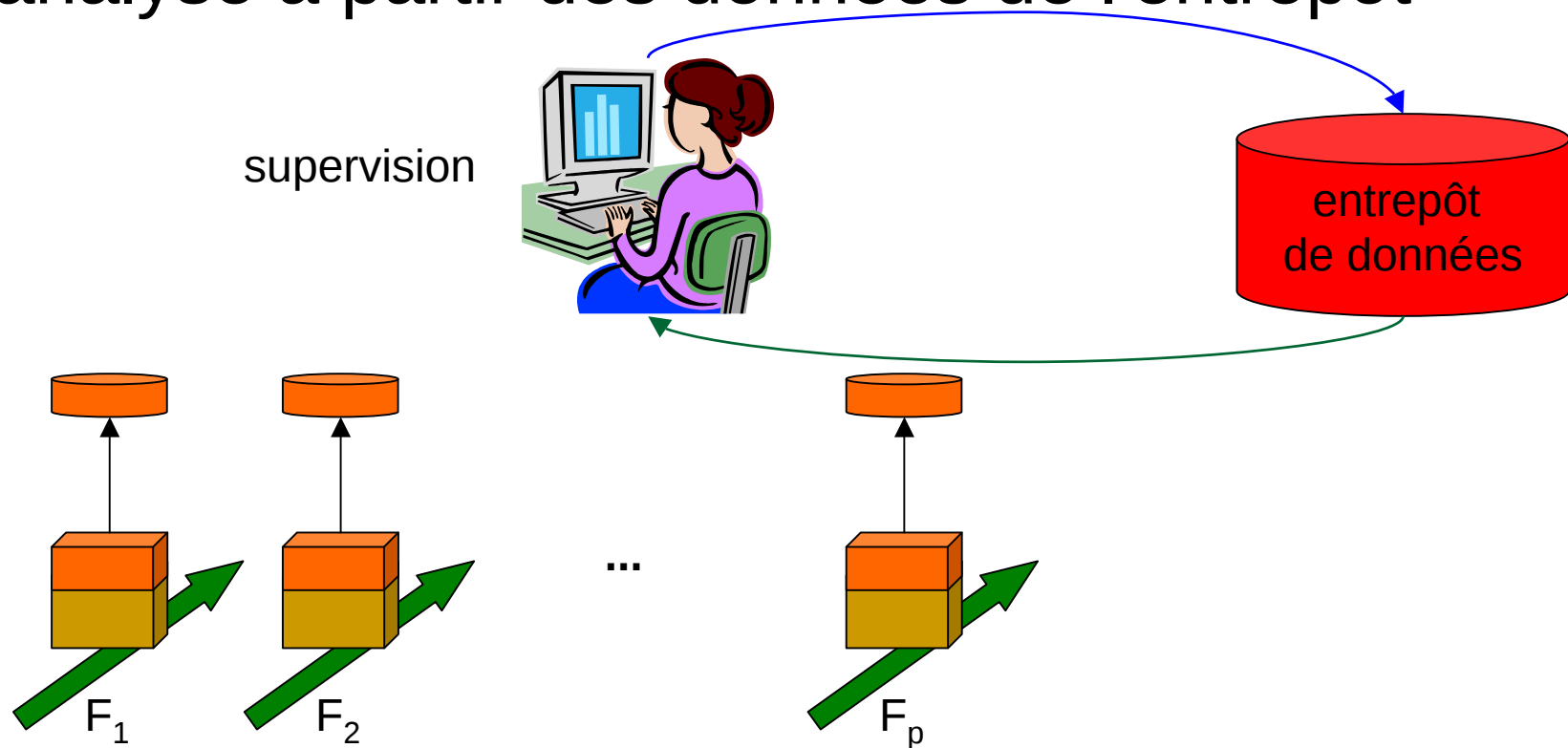
flux de données et système d'information

- chargement périodique dans le SI



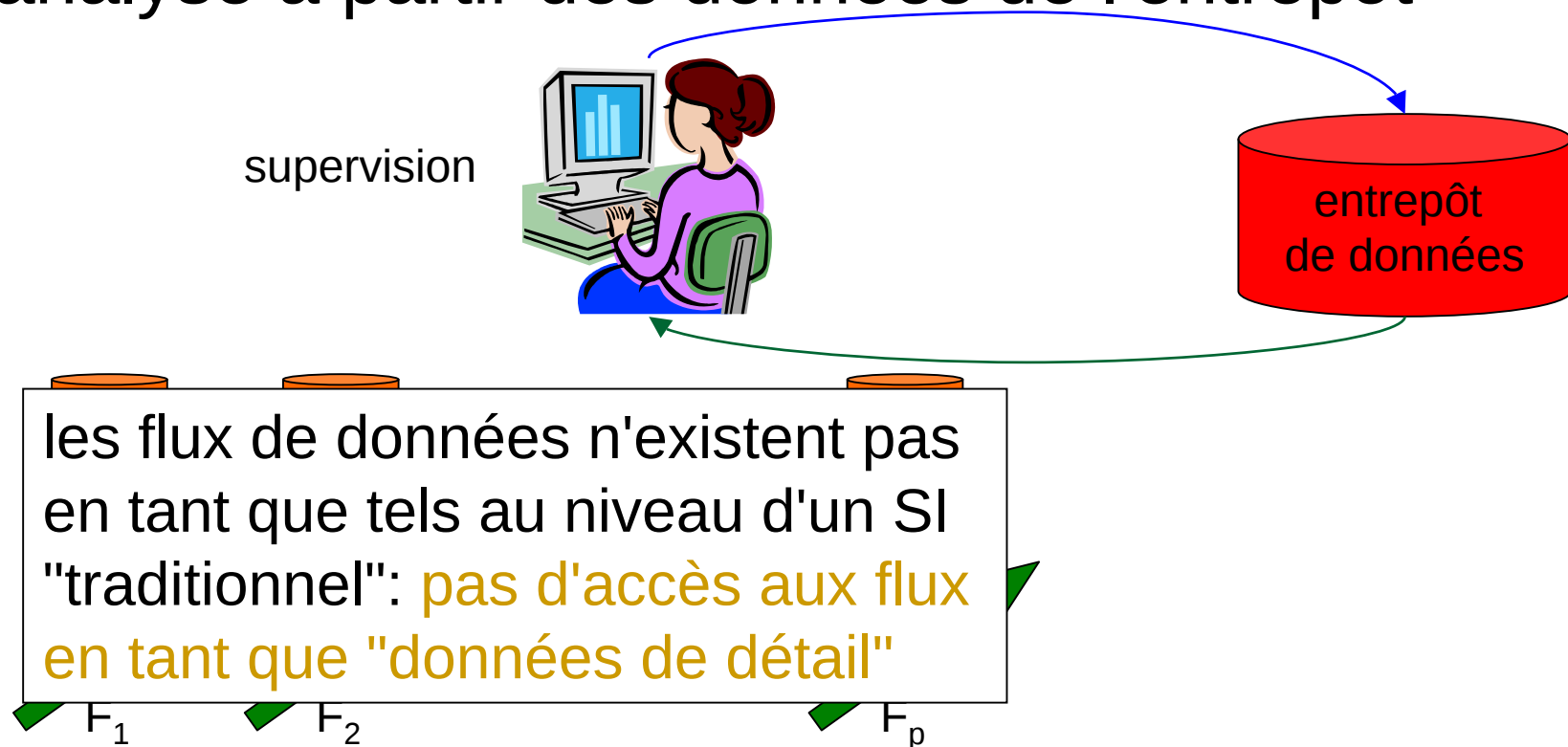
flux de données et système d'information

- analyse à partir des données de l'entrepôt



flux de données et système d'information

- analyse à partir des données de l'entrepôt

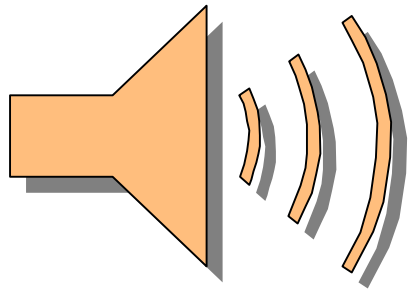


flux de données et système d'information

- défauts de l'architecture actuelle:
 - ❑ le traitement des flux est délégué à des systèmes propriétaires
 - ❑ les agrégats disponibles sont prédéfinis et le flux en tant que donnée de détail est inaccessible
 - ❑ le rafraîchissement des données de l'entrepôt est périodique
 - ❑ l'entrepôt de données est le goulot d'étranglement du système
- supervision de réseau: pas assez de réactivité
 - ❑ pour la détection des attaques type D-DoS
 - ❑ pour la supervision du trafic en temps réel
 - ❑ pour la vérification des contrats (SLA) entre opérateurs

flux de données et système d'information

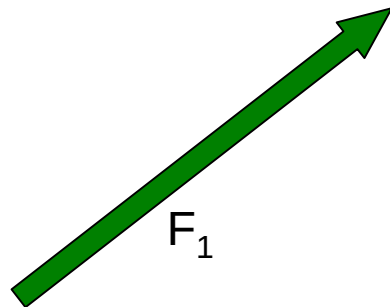
- architecture cible: servir différents clients



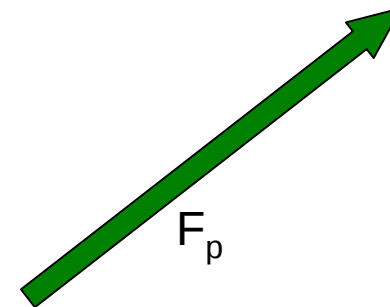
surveillance temps réel



supervision

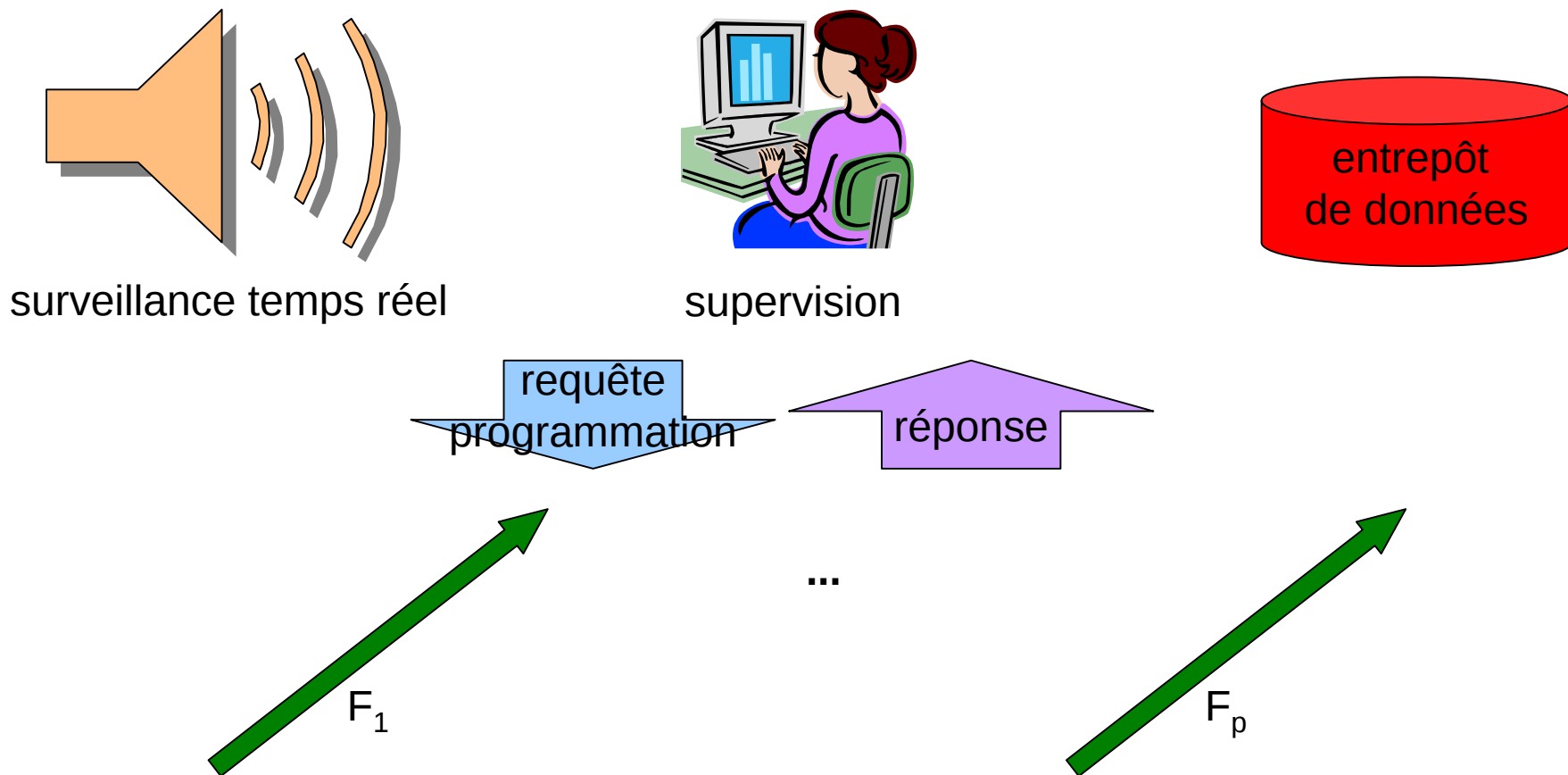


...



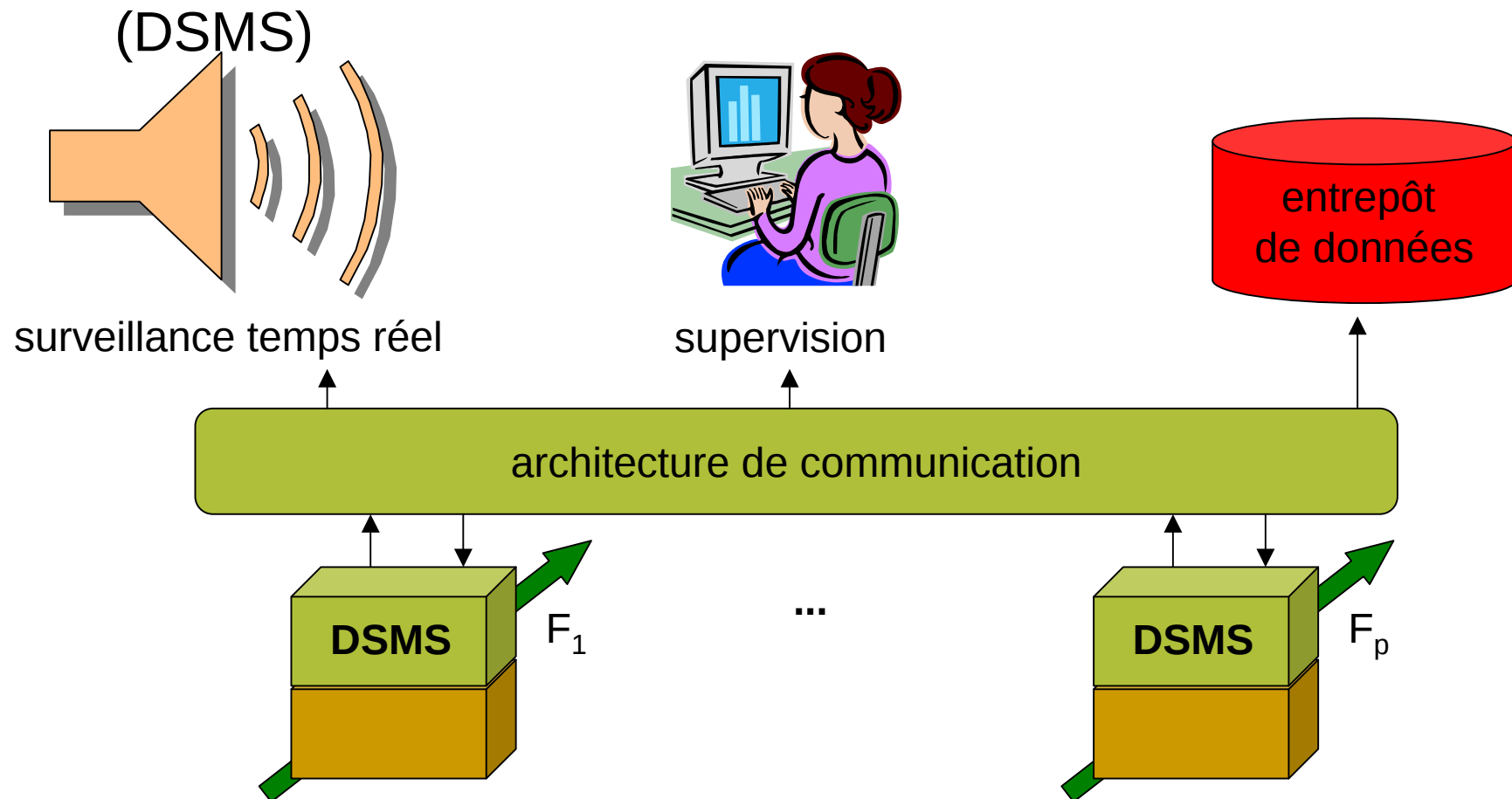
flux de données et système d'information

- architecture cible: système programmable et autonome



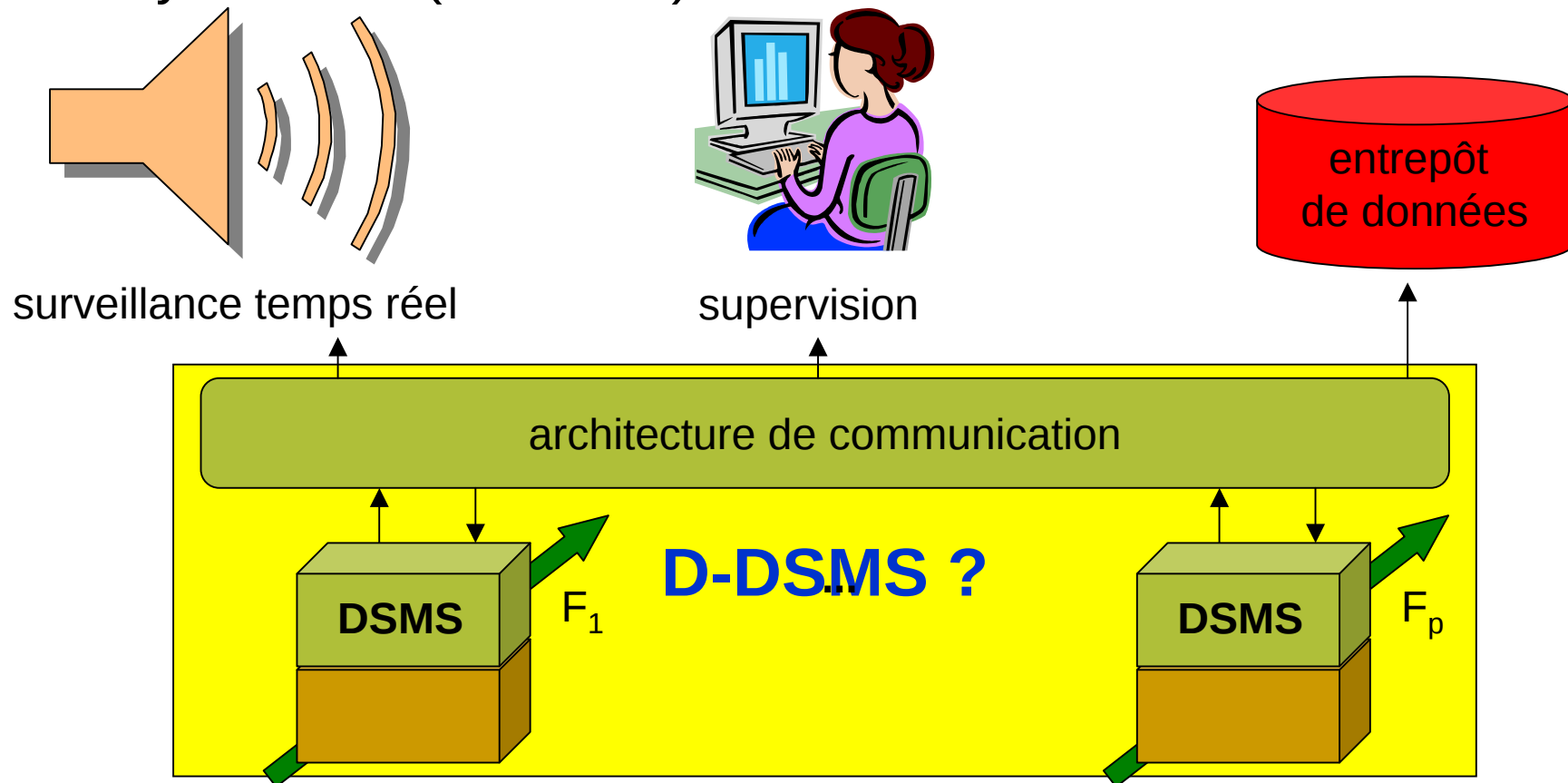
flux de données et système d'information

- architecture cible: Data Stream Management System

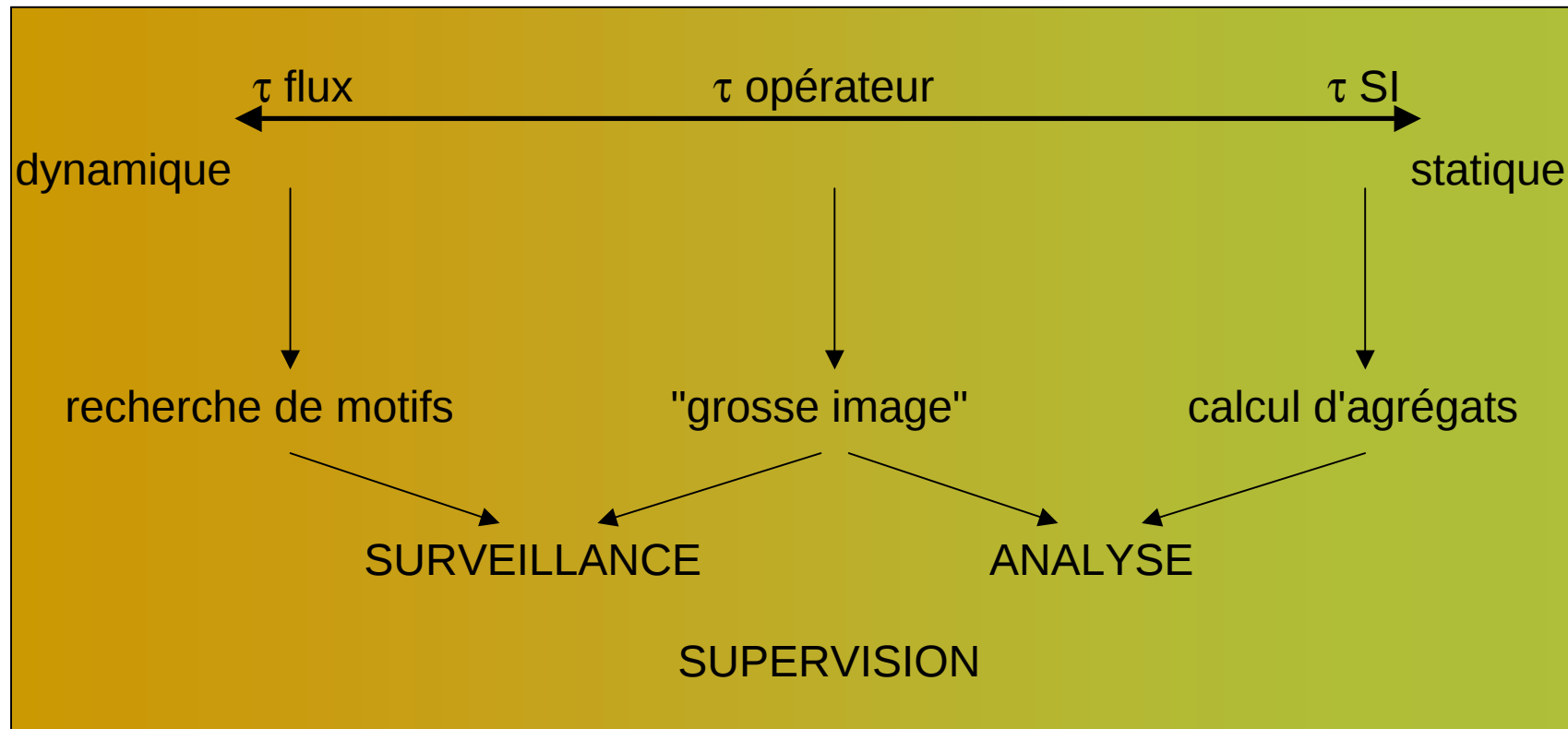


flux de données et système d'information

- architecture cible: Distributed Data Stream Management System (D-DSMS)



objectifs du traitement des flux



ystème d'information et flux de données

- deux exemples de flux de données
- définition d'un flux de données: observation et modélisation
- flux de données et système d'information
- contraintes caractéristiques du traitement des flux de données

caractéristiques du traitement des flux

- mémoire
- temps
- espace

caractéristiques du traitement des flux

- mémoire: volume infini du flux
 - utilisation efficace des capacités de stockage
 - écart entre le volume du flux et les capacités de stockage
 - exemple des capteurs: flux faible mais ressources plus faibles encore !
 - traitement en une passe

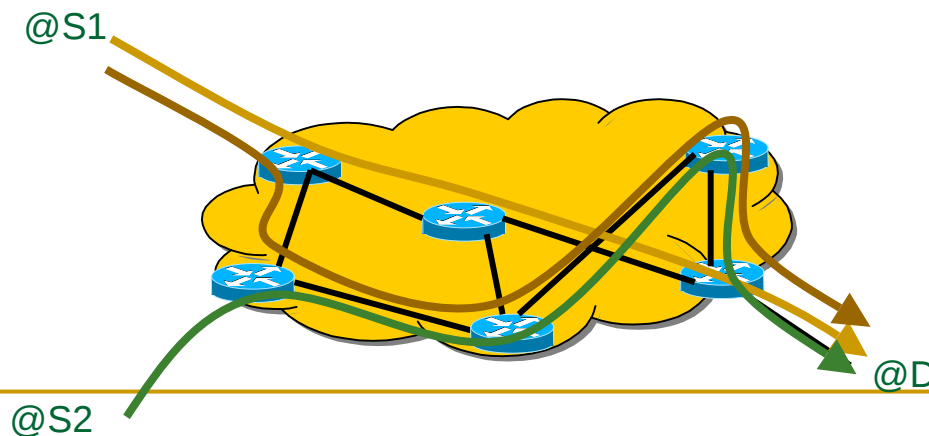
caractéristiques du traitement des flux

- temps: dynamique du flux très rapide
 - utilisation efficace des capacités de traitement
 - masquage de la dynamique du flux pour les applications clientes du DSMS

caractéristiques du traitement des flux

- espace:

- problème souvent intrinsèquement distribué
 - on ne peut pas forcément déplacer les flux vers le point de contrôle
- exemple pour un centre de supervision:
 - reconstituer par @S les volumes vers une @D donnée



caractéristiques du traitement des flux

■ espace:

- ❑ problème souvent intrinsèquement distribué
 - on ne peut pas déplacer les flux vers le point de contrôle
- ❑ requête continue vs requête ponctuelle
 - "push" d'information en continu depuis les points d'observation vers le point de contrôle
- ❑ utilisation efficace des ressources finies de communication
 - dépend de l'infrastructure de communication et de ses caractéristiques

conséquences sur le traitement des flux

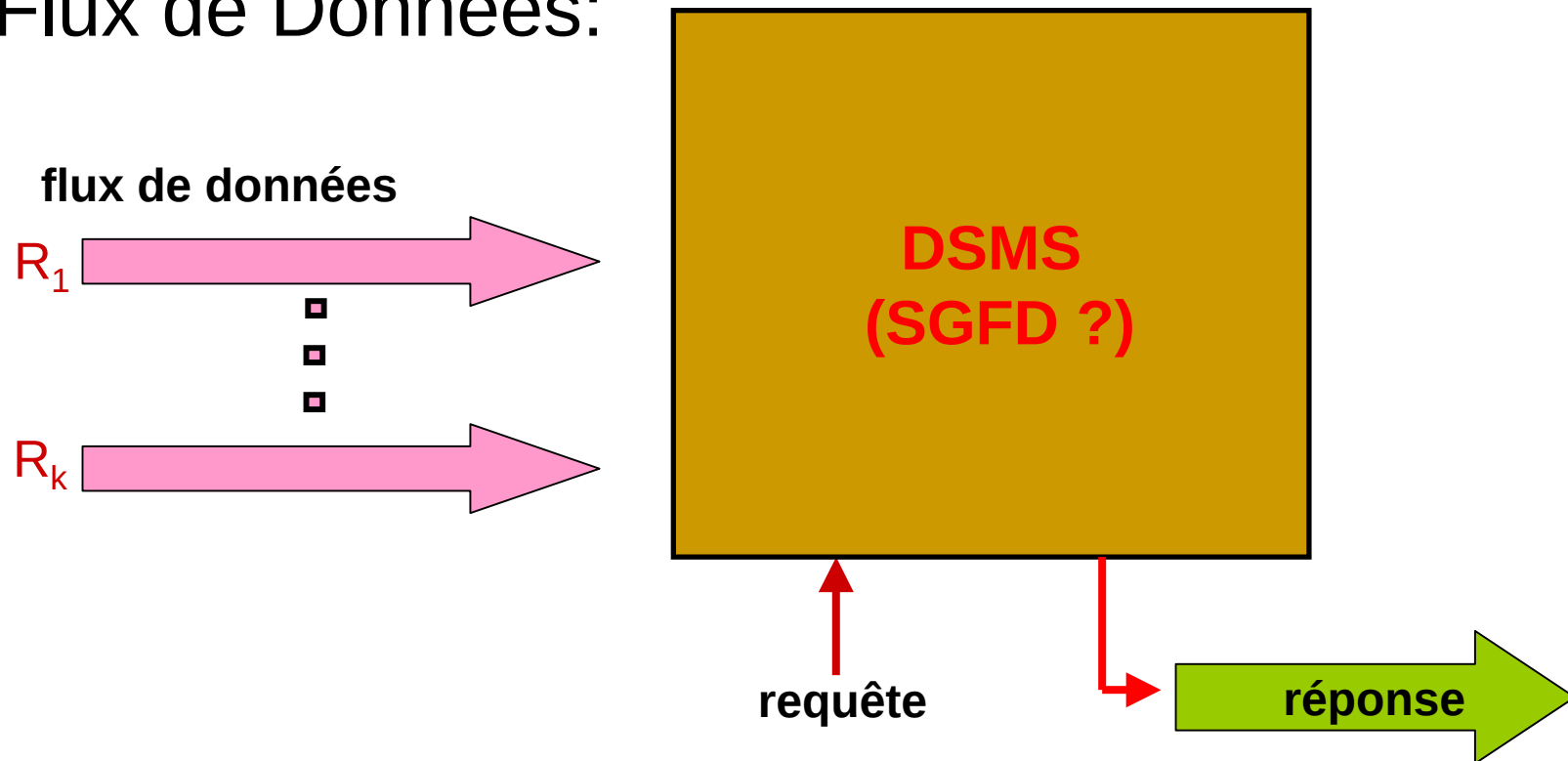
- problème fortement contraint sur trois dimensions essentielles
 - sur un point d'observation:
 - requête "holistique" vs requête "incrémentale"
 - incrémentale: max, moyenne
 - holistique:
 - médiane, nombre de valeurs différentes
 - précision vs économie de stockage et temps de calcul
 - dans le cadre distribué:
 - économie de communication → délai dans les mises à jour
 - précision vs économie de communication, même pour les requêtes incrémentales

conséquences sur le traitement des flux

- compromis performance/ressources inévitable
- techniques d'approximation indispensables
- la maîtrise du compromis passe par l'établissement de bornes sur les techniques d'approximation
 - $\Pr(|n-\tilde{n}| < \epsilon) > \delta$ avec (ϵ, δ) fonction des ressources

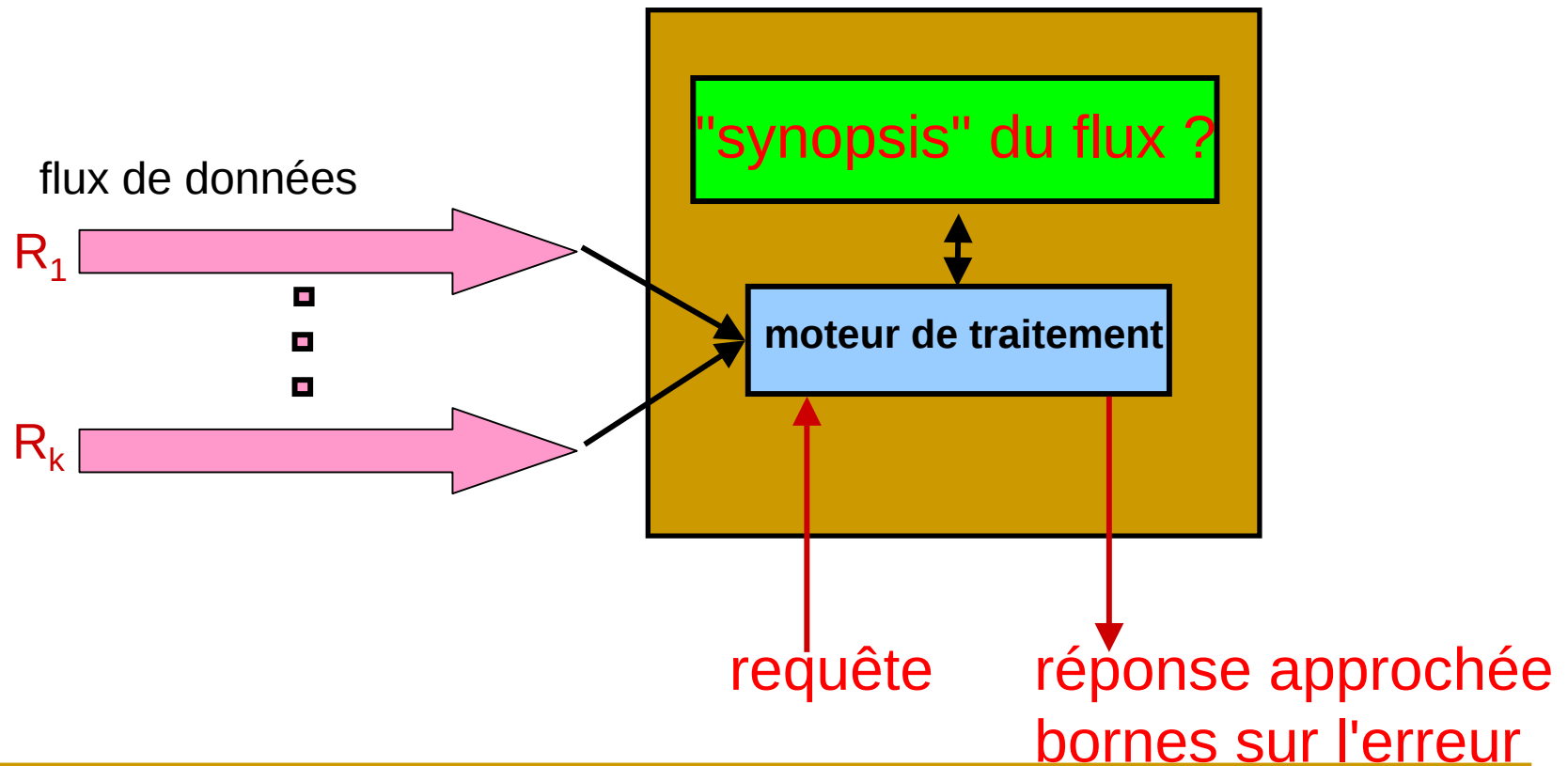
à suivre

- présentation des Systèmes de Gestion des Flux de Données:



à suivre

- techniques d'analyse des flux de données



Bibliographie

- S. Muthukrishnan, **Data Streams: Algorithms and Applications**, **Foundations and Trends in Theoretical Computer Science**, Volume 1, Issue 2, now Publishers Inc. P0 Box 1024 Hanover, MA

Plan

- Système d'information et flux de données
 - Fabrice CLEROT
- Systèmes de gestion des flux de données
 - Georges HEBRAIL
- Techniques d'analyse de flux de données
 - en un point d'observation, Baptiste CSERNE
 - distribuées, Fabrice CLEROT

Plan (partie DSMS)

- Les DSMS vus de l'utilisateur
 - □ Définition d'un DSMS
 - Modèle de données des DSMS
 - Requêtes dans un DSMS
 - Exemple STREAM « Linear Road »
 - Exemple TelegraphCQ « Conso Elec »
- Le cœur d'un DSMS
 - Architecture des DSMS
 - Réponses approchées aux requêtes
- Principaux DSMS existants
- Conclusion
- Bibliographie

Définition d'un DSMS

■ SGBD - DBMS

- ❑ **Système de Gestion de Bases de Données** - *Data Base Management System*
- ❑ Un SGBD est un logiciel permettant de gérer facilement des **bases de données**.

Une **base de données** est un ensemble d'informations :

- persistantes
- décrites dans un **modèle** indépendamment de leur utilisation
- manipulables **seulement** au travers du modèle (consultation, mise à jour)
- cohérentes par rapport au modèle défini
- partageables entre plusieurs utilisateurs
- éventuellement distribuées sur plusieurs machines
- protégées (accès, pannes)

et tout ceci avec de bonnes performances

■ SGFD - DSMS

- ❑ **Système de Gestion de Flux de Données** - *Data **Stream** Management System*
- ❑ Un SGFD est un logiciel permettant de gérer facilement des **flux de données** ?

- ❑ **Pas de définition figée aujourd'hui mais un certain consensus :**

- Gérer **plusieurs flux** de données **structurés en complément** de données relationnelles
- Permettre la définition et l'exécution (longue) de plusieurs **requêtes continues**
- Avec des ressources **limitées** en terme de capacité de **calcul** et de **mémorisation**

et tout ceci avec de bonnes performances ...

Plan (partie DSMS)

- Les DSMS vus de l'utilisateur
 - Définition d'un DSMS
 - □ **Modèle de données des DSMS**
 - Requêtes dans un DSMS
 - Exemple STREAM « Linear Road »
 - Exemple TelegraphCQ « Conso Elec »
- Le cœur d'un DSMS
 - Architecture des DSMS
 - Réponses approchées aux requêtes
- Principaux DSMS existants
- Conclusion
- Bibliographie

Modèle de données des DSMS

- Structure de relation (table)
 - N-uplet (ligne), Attribut (colonne)

CLIENT

ID_CLIENT	NOM	PRENOM	ADRESSE	VILLE
1	Dupont	Jacques	25, Rue de Paris	Bagneux
2	Duval	Pierre	12, Bd Jaurès	Orsay
3	Vincent	Isabelle	-	Paris
4	Firin	Laure	34, Rue Irun	Vélizy

- Structure de flux de données
 - N-uplet (ligne), Attribut (colonne)

« *Flux de n-uplets* »

TRAFIC_IP

Source	Destination	Duration	Bytes	Protocol
...	...	...	...	...
10.1.0.2	16.2.3.7	12	20K	http
18.6.7.1	12.4.0.3	16	24K	http
12.4.3.8	14.8.7.4	26	58K	http
19.7.1.2	16.5.5.8	18	80K	ftp
...	...	...	...	...

(Tableau adapté de Garofalakis, Gehrke, Rastogi, Tutorial VLDB 02)

Modèle de données des DSMS (2)

■ Représentation du temps dans un flux de données

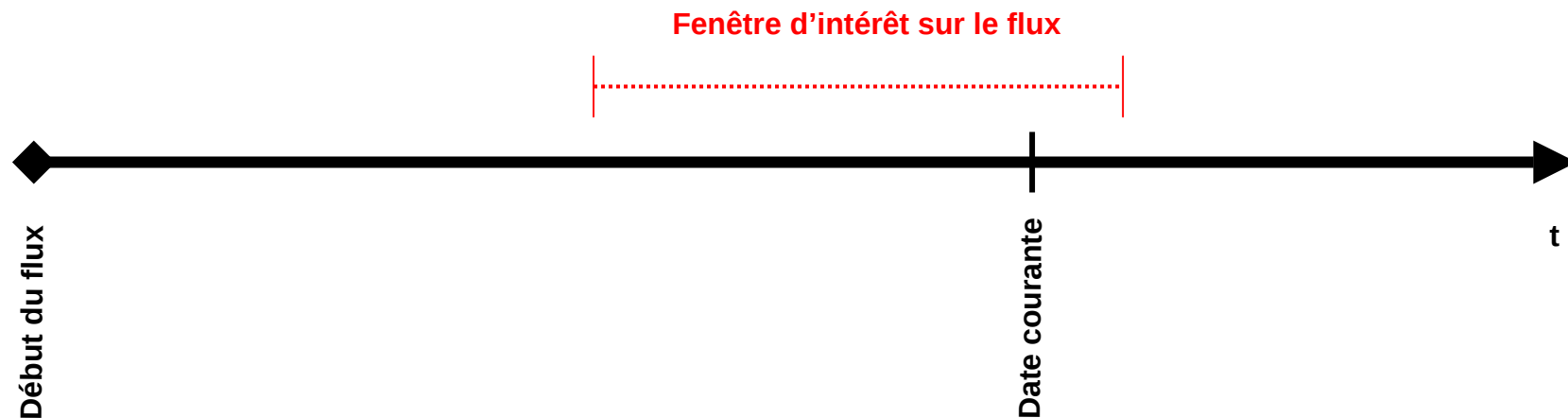
- Temps **logique** : numéro d'ordre
- Temps **physique** : date (année, mois, jour, heure, minute, seconde)
- Datation **implicite** (timestamp attribué par le DSMS à l'entrée)
- Datation **explicite** (attribut de type date dans la structure du flux)
- Ordre d'arrivée des n-uplets dans le DSMS

Timestamp	Source	Destination	Duration	Bytes	Protocol
...	...	...	...	...	...
12342	10.1.0.2	16.2.3.7	12	20K	http
12343	18.6.7.1	12.4.0.3	16	24K	http
12344	12.4.3.8	14.8.7.4	26	58K	http
12345	19.7.1.2	16.5.5.8	18	80K	ftp
...	...	...	...	...	...

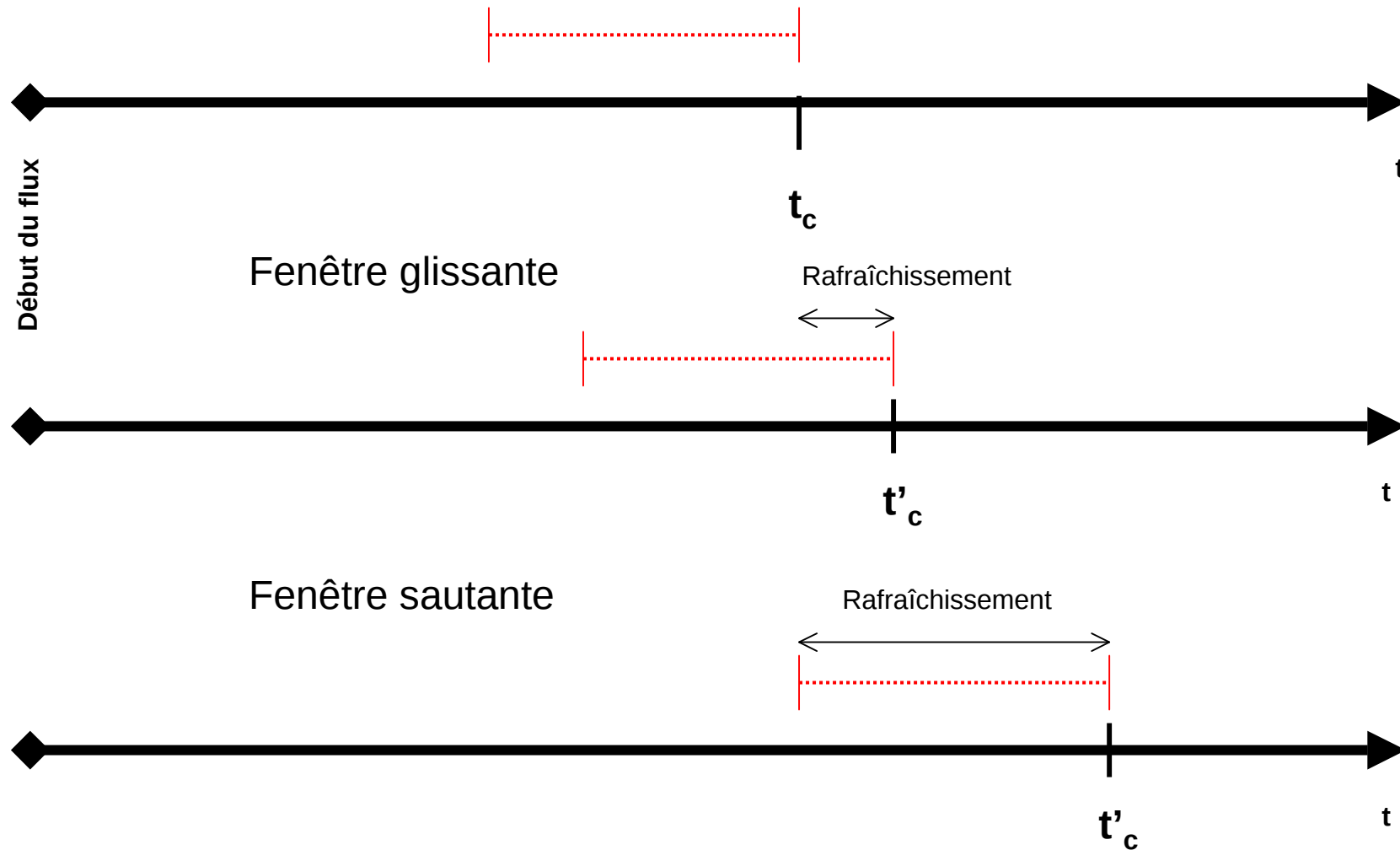
DATE-HEURE	Puis. A (kW)	Puis. R (kVAR)	U 1 (V)	I 1 (A)
...	...	...	...	...
16/12/2006-17:26	5,374	0,498	233,29	23
16/12/2006-17:27	5,388	0,502	233,74	23
16/12/2006-17:28	3,666	0,528	235,68	15,8
16/12/2006-17:29	3,52	0,522	235,02	15
...	...	...	...	...

Modèle de données des DSMS (3)

- Notion de fenêtrage sur un flux de données
 - Définition **logique** (timestamp logique : numéro, nombre de n-uplets)
 - Définition **physique** : (timestamp physique : date, durée)
 - Borne **point de repère** (landmark)
 - Borne **glissante** (sliding)
 - Intervalle de **rafraîchissement**



Modèle de données des DSMS (4)



Modèle de données des DSMS (5)

- Entrées d'un DSMS

- Relations classiques, par exemple :
 - Correspondance compteur, client, ville
 - Consommations normales heure par heure
- Un ou plusieurs flux de données, par exemple :
 - Flux de consommation d'électricité (plusieurs compteurs)
 - Flux des températures extérieures par région

- Sorties d'un DSMS

- Relations persistantes mises à jour « au fil de l'eau », par exemple :
 - Consommation électrique agrégée par ville et par heure sur les 24 dernières heures
- Un ou plusieurs flux de données, par exemple :
 - Alarme sur les clients ayant eu une consommation « anormale » sur les 24 dernières heures
 - 10 clients les plus consommateurs sur les 24 dernières heures, émis toutes les heures

Plan (partie DSMS)

- Les DSMS vus de l'utilisateur
 - Définition d'un DSMS
 - Modèle de données des DSMS
 - □ **Requêtes dans un DSMS**
 - Exemple STREAM « Linear Road »
 - Exemple TelegraphCQ « Conso Elec »
- Le cœur d'un DSMS
 - Architecture des DSMS
 - Réponses approchées aux requêtes
- Principaux DSMS existants
- Conclusion
- Bibliographie

Requêtes dans un DSMS

- Notion de requête continue
 - **Requête standard** dans un SGBD (*one-time query*)
 - Définie et exécutée une fois sur les données présentes dans la base
 - Les données sont persistantes et les requêtes éphémères
 - L'exécution d'une requête déclenche un accès aux données
 - Une fois le résultat produit, la requête est terminée
 - **Requête continue** (*continuous query*)
 - Définie et exécutée de façon **continue** avant que les données des flux ne soient arrivées
 - Les données sont éphémères et les requêtes sont persistantes
 - L'arrivée d'une donnée d'un flux déclenche un accès aux requêtes définies
 - Les requêtes continues produisent des résultats de façon continue :
 - Sous forme de flux sortants
 - Sous forme de mises à jour de relations persistantes
 - Exemples de requêtes continues
 - Consommation électrique agrégée par ville et par heure sur les 24 dernières heures
 - Clients ayant eu une consommation « anormale » sur les 24 dernières heures
 - 10 clients les plus consommateurs sur les 24 dernières heures

Requêtes dans un DSMS (2)

- Sémantique des requêtes continues
 - L'exécution des requêtes continues doit pouvoir se faire de façon **incrémentale** :
 - *$A(Q, t+1)$ s'obtient à partir de $A(Q, t)$ et des nouveaux n -uplets arrivés entre t et $t+1$ dans les flux intervenant dans la requête*
- Principaux langages de requêtes continues
 - Extensions de SQL aux requêtes continues (STREAM, TelegraphCQ)
 - Assemblage par l'utilisateur d'opérateurs s'appliquant à des flux (Aurora, Borealis)
- Extensions de SQL aux requêtes continues
 - Interroger en SQL les flux comme les relations
 - Requête sur l'ensemble d'un ou plusieurs flux :
 - Problème des « opérations bloquantes » qui nécessitent d'avoir vu tout le flux avant de produire le résultat. Exemple : tri, agrégats.
 - Requête avec spécification des fenêtres temporelles sur les flux :
 - Une solution aux opérations bloquantes
 - Rend incrémentale l'exécution des requêtes continues
 - Ponctuation

Requêtes dans un DSMS (3)

- Quelques exemples de requêtes continues
 - CONSO (**ID_COMPTEUR**, JOUR, HEURE, MINUTE, ENERGIE_KWH)

ID_COMPTEUR	JOUR	HEURE	MINUTE	ENERGIE_KWH
...	...	...	...	...
3214324	16/12/2006	17	26	0,374
5435453	16/12/2006	17	27	0,388
7576744	16/12/2006	17	28	0,666
2452543	16/12/2006	17	26	0,52
...	...	...	...	...

Requêtes dans un DSMS (4)

- Quelques exemples de requêtes continues (*en pseudo CQL*)
 - **CONSO (ID_COMPTEUR, JOUR, HEURE, MINUTE, ENERGIE_KWH)**
 - Select * from CONSO where ENERGIE_KWH > 20 ;
Select * from CONSO [**RANGE 5 HOURS**] where ENERGIE_KWH > 20 ;
 - Select distinct ID_COMPTEUR from CONSO ;
Select distinct ID_COMPTEUR from CONSO [**RANGE 5 HOURS**] ;
 - Select ID_COMPTEUR, JOUR, SUM(ENERGIE_KWH)
From CONSO
Group by ID_COMPTEUR, JOUR ;
 - Select VILLE, JOUR, SUM(ENERGIE_KWH)
From CONSO CO, CLIENT CL
Where CO.ID_COMPTEUR = CL.ID_COMPTEUR
Group by VILLE, JOUR ;
 - Emission d'un flux versus maintenance d'une relation
 - Fenêtrage versus ponctuation

Requêtes dans un DSMS (5)

- Quelques exemples de requêtes continues (*en pseudo CQL*)
 - **CONSO_ELEC** (ID_COMPTEUR, JOUR, HEURE, MINUTE, ENERGIE_KWH)
 - **CONSO_GAZ** (ID_COMPTEUR, JOUR, HEURE, MINUTE, ENERGIE_KWH)
 - ```
Select C1.JOUR , C1.HEURE, C1.MINUTE,
 (C1. ENERGIE_KWH + C2. ENERGIE_KWH) * 0,1 PRIX
From CONSO_ELEC C1, CONSO_GAZ C2
Where C1.JOUR = C2.JOUR AND C1.HEURE = C2.HEURE AND C1.MINUTE = C2.MINUTE
 AND C1.ID_COMPTEUR = C2.ID_COMPTEUR ;
```
    - ```
Select C1.JOUR , C1.HEURE, C1.MINUTE,  
      ( C1. ENERGIE_KWH + C2. ENERGIE_KWH ) * 0,1 PRIX  
From CONSO_ELEC C1 [RANGE 10 MINUTES], CONSO_GAZ C2 [RANGE 10 MINUTES],  
Where C1.JOUR = C2.JOUR AND C1.HEURE = C2.HEURE AND C1.MINUTE = C2.MINUTE  
      AND C1.ID_COMPTEUR = C2.ID_COMPTEUR ;
```
- Emission d'un flux versus maintenance d'une relation
- Fenêtrage versus ponctuation

Plan (partie DSMS)

- Les DSMS vus de l'utilisateur
 - Définition d'un DSMS
 - Modèle de données des DSMS
 - Requêtes dans un DSMS
 - □ Exemple STREAM « Linear Road »
 - Exemple TelegraphCQ « Conso Elec »
- Le cœur d'un DSMS
 - Architecture des DSMS
 - Réponses approchées aux requêtes
- Principaux DSMS existants
- Conclusion
- Bibliographie

Plan (partie DSMS)

- Les DSMS vus de l'utilisateur
 - Définition d'un DSMS
 - Modèle de données des DSMS
 - Requêtes dans un DSMS
 - Exemple STREAM « Linear Road »
 - □ Exemple TelegraphCQ « Conso Elec »
- Le cœur d'un DSMS
 - Architecture des DSMS
 - Réponses approchées aux requêtes
- Principaux DSMS existants
- Conclusion
- Bibliographie

Exemple TelegraphCQ Conso Elec

■ TelegraphCQ Version 2.1

- Université de Berkeley
- DSMS généraliste
- Langage de type CQL (Continuous Query Language) déclaratif basé sur SQL
- Extension du SGBD PostgreSQL (mode d'exécution particulier)
- Structures de données :
 - Structure relationnelle de PostgreSQL
 - Structure de flux (CREATE STREAM ...)
- Alimentation des flux par des ports TCP/IP associés à des wrappers
- Gestion physique du temps (timestamp à prévoir dans les n-uplets des flux)
- Clauses de fenêtrage (physique, glissantes, point de repère, sautantes)
- Possibilité **d'ajouter dynamiquement des requêtes**
- Exécution partagée et adaptation dynamique des plans d'exécution des requêtes
- Toujours en évolution et développement

Exemple TelegraphCQ Conso Elec (2)

- Données de consommation électrique
 - Flux d'entrée : index de plusieurs compteurs observés toutes les 2 secondes

```
CREATE STREAM Elec.flux  
  ( compteur varchar(12), index INTEGER, tcqtime TIMESTAMP TIMESTAMPCOLUMN )  
  TYPE ARCHIVED ;  
ALTER STREAM Elec.flux ADD WRAPPER csvwrapper ;
```

compteur	index	tcqtime
...	...	...
0501260611XX	11089624	2003-12-04 07:53:02
105414921542	11064549	2003-12-04 07:53:02
163816431661	11012977	2003-12-04 07:53:02
0501260611XX	11089624	2003-12-04 07:53:04
105414921542	11064556	2003-12-04 07:53:04
163816431661	11012989	2003-12-04 07:53:04
0501260611XX	11089624	2003-12-04 07:53:06
105414921542	11064567	2003-12-04 07:53:06
163816431661	11013001	2003-12-04 07:53:06
0501260611XX	11089624	2003-12-04 07:53:08
...	...	...

Exemple TelegraphCQ Conso Elec (3)

- Consommation des cinq dernières minutes, minute par minute, par compteur

	compteur	minindex	tcqtime	minindex	consommation	tcqtime
	-----+-----+-----+-----+-----+-----+-----					
→	0501260611XX	11089624	2003-12-04 07:53:00	11089626	2	2003-12-04 07:54:00
→	0501260611XX	11089624	2003-12-04 07:53:00	11089626	2	2003-12-04 07:54:00
	0501260611XX	11089626	2003-12-04 07:54:00	11089696	70	2003-12-04 07:55:00
→	0501260611XX	11089624	2003-12-04 07:53:00	11089626	2	2003-12-04 07:54:00
	0501260611XX	11089626	2003-12-04 07:54:00	11089696	70	2003-12-04 07:55:00
	0501260611XX	11089696	2003-12-04 07:55:00	11089767	71	2003-12-04 07:56:00
→	0501260611XX	11089624	2003-12-04 07:53:00	11089626	2	2003-12-04 07:54:00
	0501260611XX	11089626	2003-12-04 07:54:00	11089696	70	2003-12-04 07:55:00
	0501260611XX	11089696	2003-12-04 07:55:00	11089767	71	2003-12-04 07:56:00
	0501260611XX	11089767	2003-12-04 07:56:00	11089836	69	2003-12-04 07:57:00
→	0501260611XX	11089624	2003-12-04 07:53:00	11089626	2	2003-12-04 07:54:00
	0501260611XX	11089626	2003-12-04 07:54:00	11089696	70	2003-12-04 07:55:00
	0501260611XX	11089696	2003-12-04 07:55:00	11089767	71	2003-12-04 07:56:00
	0501260611XX	11089767	2003-12-04 07:56:00	11089836	69	2003-12-04 07:57:00
	0501260611XX	11089836	2003-12-04 07:57:00	11089907	71	2003-12-04 07:58:00
→	0501260611XX	11089626	2003-12-04 07:54:00	11089696	70	2003-12-04 07:55:00
	0501260611XX	11089696	2003-12-04 07:55:00	11089767	71	2003-12-04 07:56:00
	0501260611XX	11089767	2003-12-04 07:56:00	11089836	69	2003-12-04 07:57:00
	0501260611XX	11089836	2003-12-04 07:57:00	11089907	71	2003-12-04 07:58:00
	0501260611XX	11089907	2003-12-04 07:58:00	11089975	68	2003-12-04 07:59:00

Exemple TelegraphCQ Conso Elec (4)

- Consommation des cinq dernières minutes, minute par minute, par compteur

```
CREATE STREAM Elec.minflux1 ( compteur varchar(12), minindex INTEGER, tcqtime TIMESTAMP
    TIMESTAMPCOLUMN ) TYPE UNARCHIVED ;
CREATE STREAM Elec.minflux2 ( identique à Elec.minflux1 ) ;
WITH
Elec.minflux1 AS (
    SELECT compteur,min(index),DATE_TRUNC('minute', tcqtime)
    FROM Elec.flux [RANGE BY '6 minutes' SLIDE BY '1 minute'
    START AT '2003-12-04 07 :50 :00']
    GROUP BY compteur, DATE_TRUNC('minute', tcqtime) )
Elec.minflux2 AS (
    SELECT ... même définition que Elec.minflux1 avec RANGE BY à 5 minutes )
(
    SELECT f1.compteur, f1.minindex, f1.tcqtime, f2.minindex, f2.minindex-f1.minindex, f2.tcqtime
    FROM Elec.minflux1 as f1 [RANGE BY '1 minute' SLIDE BY '1 minute' START AT '2003-12-04 07 :50 :00'],
        Elec.minflux2 as f2 [RANGE BY '1 minutes' SLIDE BY '1 minute' START AT '2003-12-04 07 :50 :00']
    WHERE f1.compteur = f2.compteur AND f1.tcqtime = (f2.tcqtime - interval '1 minute')) ;
```

Exemple TelegraphCQ Conso Elec (5)

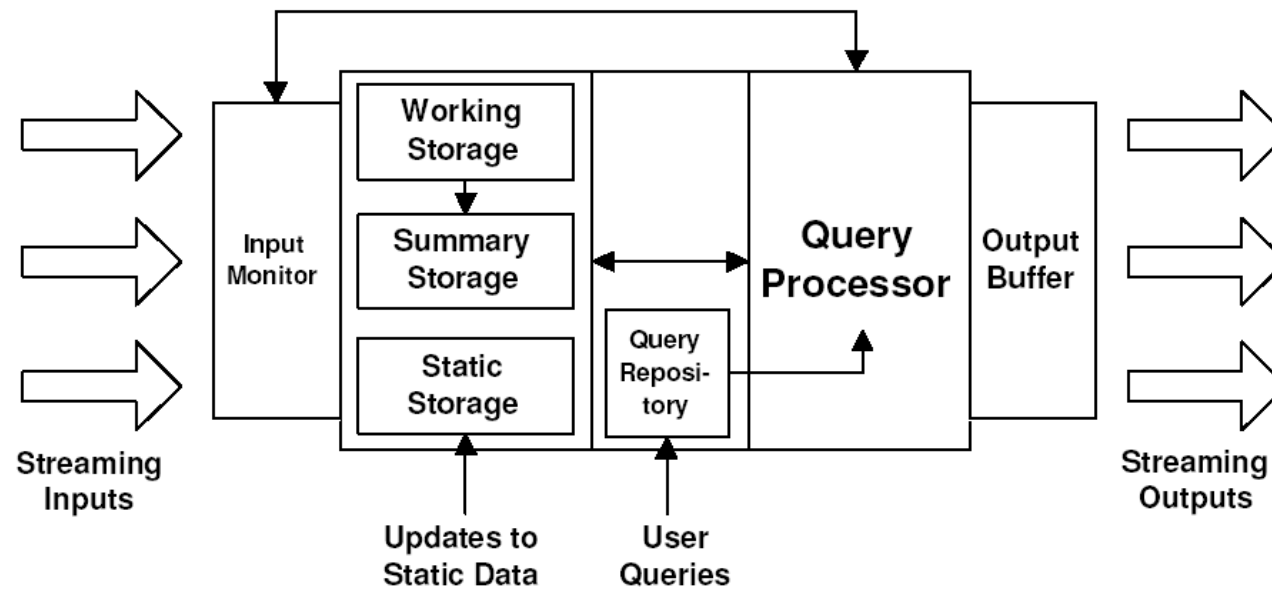
- ❑ Consommation des cinq dernières minutes, minute par minute, par ville
 - ❑ Jointure avec une table CompteurVille
- ❑ Alarme de consommation anormale
 - ❑ Table de consommation horaire normale à 20°
 - ❑ Flux des températures extérieures
 - ❑ Calcul des consommations horaires normales et observées
 - ❑ Émission d'une alarme à partir de 12h

Plan (partie DSMS)

- Les DSMS vus de l'utilisateur
 - Définition d'un DSMS
 - Modèle de données des DSMS
 - Requêtes dans un DSMS
 - Exemple STREAM « Linear Road »
 - Exemple TelegraphCQ « Conso Elec »
- Le cœur d'un DSMS
- □ **Architecture des DSMS**
- Réponses approchées aux requêtes
- Principaux DSMS existants
- Conclusion
- Bibliographie

Architecture des DSMS

- Encore en travaux, très dépendant des prototypes
- Architecture générique proposée par Golab et Ozsu (2003) :



Source : Golab & Özsu 2003

Architecture des DSMS (2)

- Quelques idées directrices
 - Détermination des besoins en mémoire
 - Requête calculable avec mémoire bornée de façon exacte ou approchée
 - Génération de plans d'exécution pour un ensemble de requêtes
 - Inscription des requêtes et traitement piloté par les arrivées des flux
 - Enchaînements d'opérateurs avec des files d'attente de données (nécessité d'un 'scheduler')
 - Optimisation de la mémoire utilisée (et du temps de calcul) :
 - Partage de plans d'exécution entre plusieurs requêtes (gestion de buffers, synopsis)
 - Indexation des requêtes
 - Adaptation dynamique des plans d'exécution (requêtes longues, nouvelles requêtes)
 - Distribution des traitements (réseaux de capteurs)

Plan (partie DSMS)

- Les DSMS vus de l'utilisateur
 - Définition d'un DSMS
 - Modèle de données des DSMS
 - Requêtes dans un DSMS
 - Exemple STREAM « Linear Road »
 - Exemple TelegraphCQ « Conso Elec »
- Le cœur d'un DSMS
 - Architecture des DSMS
 - □ Réponses approchées aux requêtes
- Principaux DSMS existants
- Conclusion
- Bibliographie

Réponses approchées aux requêtes

- Quand ?
 - Requêtes nécessitant une taille mémoire non bornée
 - Ex : *10 adresses IP occupant le plus un nœud du réseau IP*
 - Flux avec un débit trop important pour le système ou exigences de délais de réponse aux requêtes
 - Atteinte de la limite de CPU
 - Atteinte de la limite de mémoire
- Solution : réponses approchées aux requêtes
 - Fenêtres glissantes sur les flux (*sliding windows*)
 - Fréquence de rafraîchissement des résultats (*batch processing*)
 - Echantillonnage (*sampling*) et techniques de 'load shedding'
 - Gestion de structures de résumés (*synopsis*)

Réponses approchées aux requêtes (2)

- Techniques de load shedding

- Objectif :

- Adapter le fonctionnement à la charge via la suppression de n-uplets (flux et requêtes)
 - Assurer une dégradation contrôlée des performances lorsque le système est saturé

- Principe :

- Placer des opérateurs d'échantillonnage dans le diagramme d'exécution du DSMS (*data flow diagram*)
 - Optimiser dynamiquement ce placement pour maintenir une qualité de service

Réponses approchées aux requêtes (3)

- Load shedding proposé par Babcock, Datar et Motwani (Equipe STREAM)
 - Requêtes d'agrégation :
 - SUM, COUNT
 - Sélections intermédiaires
 - Jointures externes par des clés étrangères avec des relations fixes

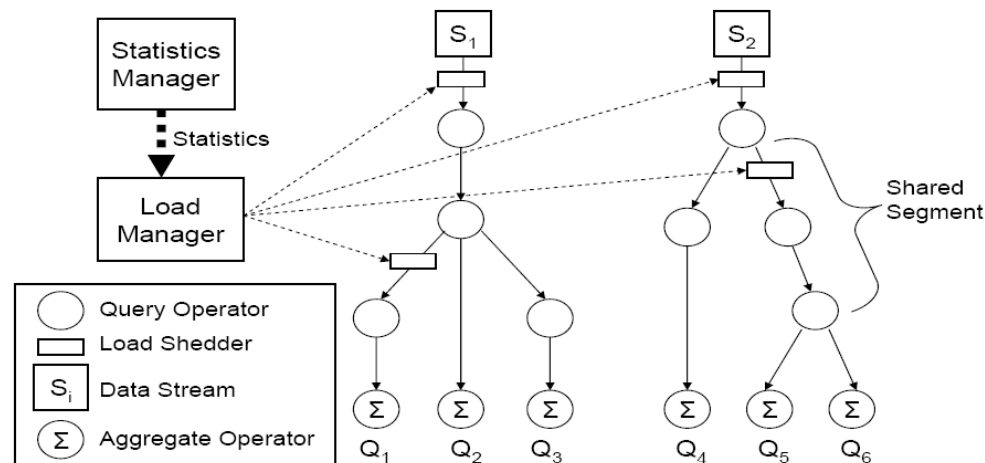


Figure 1. Data Flow Diagram

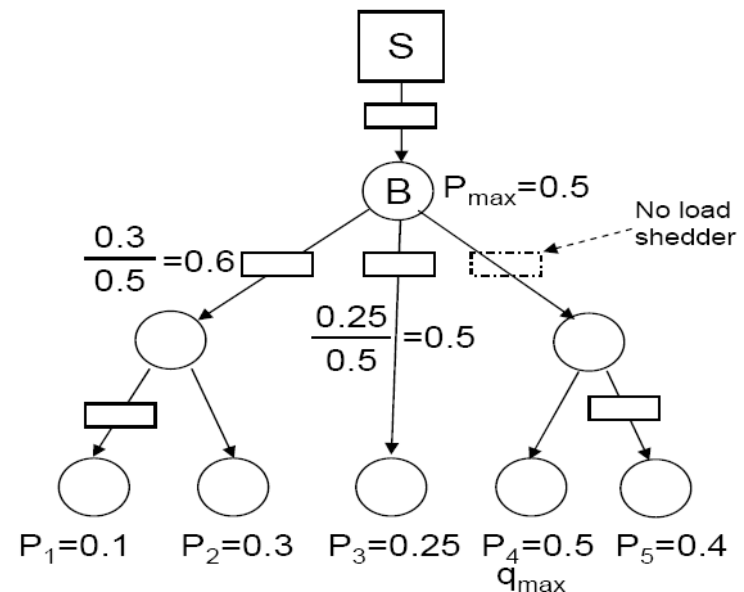
Réponses approchées aux requêtes (4)

Paramètres du problème

- Pour chaque opérateur O_i : sélectivité s_i , temps de traitement d'un n-uplet t_i
- Pour chaque opérateur terminal (SUM) : moyenne μ_i et écart type σ_i du résultat
- Pour chaque flux : r_i taux d'arrivée des n-uplets par unité de temps
- Pour chaque opérateur O_i : p_i le nombre de n-uplets par unité de temps à lui envoyer

Définition du problème :

- Déterminer les p_i minimisant l'erreur maximum sur les opérateurs terminaux sous la contrainte que la charge maximum de traitement du système ne soit pas dépassée



Réponses approchées aux requêtes (5)

- Gestion de structures de résumés (*synopsis*)
 - Exemple de motivation : les requêtes de type « **top-k** »
« *Maintenir les 10 adresses IP les plus fréquentes (dans la dernière heure ?)* »
 - Solution exacte :
 - Création et mise à jour d'un tableau des adresses observées avec comptage
(6 comptages si fréquence de rafraîchissement de 10')
 - Exécution d'une requête à la demande sur ce(s) tableau(x)
 - Problèmes :
 - Très gros tableau de taille non bornée
 - Temps de recherche et de mise à jour d'une adresse (index)
 - Solutions approchées avec synopsis :
 - Si N est le nombre d'adresses IP observables sur une période, maintenir une structure de taille bornée (ou peu croissante) permettant de répondre à la question avec une précision contrôlée
 - Maintien d'un échantillon de taille fixe (réservoir) avec ou sans fréquences
 - Utilisation de sketches – *count sketch* par exemple (voir partie *stream mining*)

Plan (partie DSMS)

- Les DSMS vus de l'utilisateur
 - Définition d'un DSMS
 - Modèle de données des DSMS
 - Requêtes dans un DSMS
 - Exemple STREAM « Linear Road »
 - Exemple TelegraphCQ « Conso Elec »
- Le cœur d'un DSMS
 - Architecture des DSMS
 - Réponses approchées aux requêtes
- ■ Principaux DSMS existants
- Conclusion
- Bibliographie

Principaux DSMS existants

- Références : Golab & Oszu 2003, Gobel & Plagemann 2004
- Principaux DSMS généralistes :
 - **STREAM** : Université de Stanford
 - Langage CQL
 - Optimisation des requêtes avec bonne gestion de la mémoire
 - Réponse approchée par gestion de synopsis
 - **TelegraphCQ** : Université de Berkeley
 - Extension de PostgreSQL
 - Requêtes continues de type CQL
 - Ajout dynamique de requêtes
 - **Aurora** (Medusa, Borealis) : Brandeis, Brown University, MIT
 - Requêtes par assemblage d'opérateurs sous forme de workflow
 - Mécanisme de load shedding avec définition de Qualité de Service
 - Medusa puis Borealis pour une distribution des traitements

Principaux DSMS existants (2)

- Principaux DSMS spécialisés :
 - **Gigascop** et **Hancock** : AT&T
 - Monitoring de réseau informatique
 - Gestion de flux de télécommunications
 - **NiagaraCQ** : University of Wisconsin-Madison
 - Requêtes continues sur du contenu web (XML-QL)
 - **Tradebot** (finance), **Statstream** (statistique)
- Réseaux de capteurs (sensor networks) :
 - **Cougar** : Cornell University
 - **TinyDB** : Université de Berkeley

Plan (partie DSMS)

- Les DSMS vus de l'utilisateur
 - Définition d'un DSMS
 - Modèle de données des DSMS
 - Requêtes dans un DSMS
 - Exemple STREAM « Linear Road »
 - Exemple TelegraphCQ « Conso Elec »
- Le cœur d'un DSMS
 - Architecture des DSMS
 - Réponses approchées aux requêtes
- Principaux DSMS existants
- ■ **Conclusion**
- Bibliographie

Conclusion (Partie DSMS)

- Outils génériques pour le traitement de flux de données
 - Cf. les BD pour le traitement de données persistantes
 - Mais une autre façon de penser les requêtes
- Supervision de systèmes gérant des flux
 - Pose de sondes au plus près des flux de **données détaillées**
 - Déclenchement d'alarmes
 - Couplage avec fouille de flux de données pour définir les comportements anormaux
- Applications déjà nombreuses
- DSMS et avenir du décisionnel
 - Coût et engorgement des entrepôts de données
 - Gestion de tableaux de bord et DSMS
 - ETL versus DSMS
 - Flexibilité du décisionnel

Conclusion (Partie DSMS)

- Maturité des DSMS ?
- Quelques challenges
 - La définition d'un langage de requête stabilisé
 - La gestion du temps dans les flux
 - L'optimisation des requêtes continues multiples
 - La gestion des requêtes approchées
 - Les DSMS distribués

Plan (partie DSMS)

- Les DSMS vus de l'utilisateur
 - Définition d'un DSMS
 - Modèle de données des DSMS
 - Requêtes dans un DSMS
 - Exemple STREAM « Linear Road »
 - Exemple TelegraphCQ « Conso Elec »
- Le cœur d'un DSMS
 - Architecture des DSMS
 - Réponses approchées aux requêtes
- Principaux DSMS existants
- Conclusion
- ■ **Bibliographie**

Bibliographie (Partie DSMS)

Data STREAM Management Systems - Applications, Concepts, and Systems. V.Goebel, T.Plagemann, Tutorial MIPS'2004, 2004.

Querying and Mining Data Streams: You Only Get One Look. A tutorial. M.Garofalakis, J.Gehrke, R.Rastogi, Tutorial SIGMOD'02, Juin 2002.

Issues in Data STREAM Management. L.Golab, M.T.Özsu, Canada. SIGMOD Record, Vol. 32, No. 2, June 2003.

Models and Issues in data stream systems. B.Babcock, S.Babu, M.Datar, R.Motwani, J.Widom, PODS'2002, Juin 2002.

STREAM: The Stanford Data STREAM Management System. A.Arasu, B.Babcock, S.Babu, J.Cieslewicz, M.Datar, K.Ito, R.Motwani, U.Srivastava, J.Widom. Department of Computer Science, Stanford University. Mars 2004. Disponible sur le site de STREAM :
<http://www-db.stanford.edu/stream>

TelegraphCQ: Continuous Dataflow Processing for an Uncertain World. S.Chandrasekaran, O.Cooper, A.Deshpande, M.J.Franklin, J.M.Hellerstein, W.Hong (Intel Berkeley Laboratory), S.Krishnamurthy, S.Madden, V.Raman (IBM Almaden Research Center), F.Reiss, M.Shah. (Université de Berkeley). CIDR 2003. <http://telegraph.cs.berkeley.edu/telegraphcq/v2.1/>

Linear Road: A Stream Data Management Benchmark. A.Arasu, M.Cherniack, E.Galvez, D.Maier, A.S.Maskey, E.Ryvkina, M.Stonebraker, R.Tibbetts, Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004. <http://www.cs.brandeis.edu/~linearroad/>

Load Shedding for Aggregation Queries over Data Streams. B.Babcock, M.Datar, R.Motwani, 2004. Disponible sur le site de STREAM.

Traitement de données de consommations électriques par un Système de Gestion de Flux de Données. T.Abdessalem, R.Chiky, G.Hébrail, J.L.Vitti, EGC'2007, Namur, 2007.

Exemple STREAM « Linear Road »

- « **Linear Road: A Stream Data Management Benchmark** », A. Arasu, M.Cherniack, E.Galvez, D.Maier, A.Maskey, E.Ryvkina, M.Stonebraker, and R.Tibbetts. *VLDB Conference*, Sept. 2004.
- Benchmark de performances de DSMS (issu de STREAM et Aurora)
- Linear City :
 - ❑ Ville fictive de 100 miles par 100 miles
 - ❑ 10 autoroutes parallèles à 3 x 3 voies + voies d'entrée/sortie
 - ❑ Les véhicules donnent leur position toutes les 30'
 - ❑ Une horloge unique, pas d'interpolation de position, pas de retard dans les données
 - ❑ Générateur aléatoire de trafic automobile avec accidents toutes les 20'
 - ❑ **Position reports (Time, VID, Spd, Xway, Lane, Dir, Seg, Pos)**
 - ❑ Requêtes des véhicules (Solde du compte, Dépenses par autoroute par jour dans les 10 dernières semaines, Estimation du temps et du prix pour un trajet sur une route donnée pour un jour de la semaine et une heure donnés)
 - ❑ Péage dépendant du trafic (augmentant avec le trafic)

Exemple STREAM « Linear Road » (2)

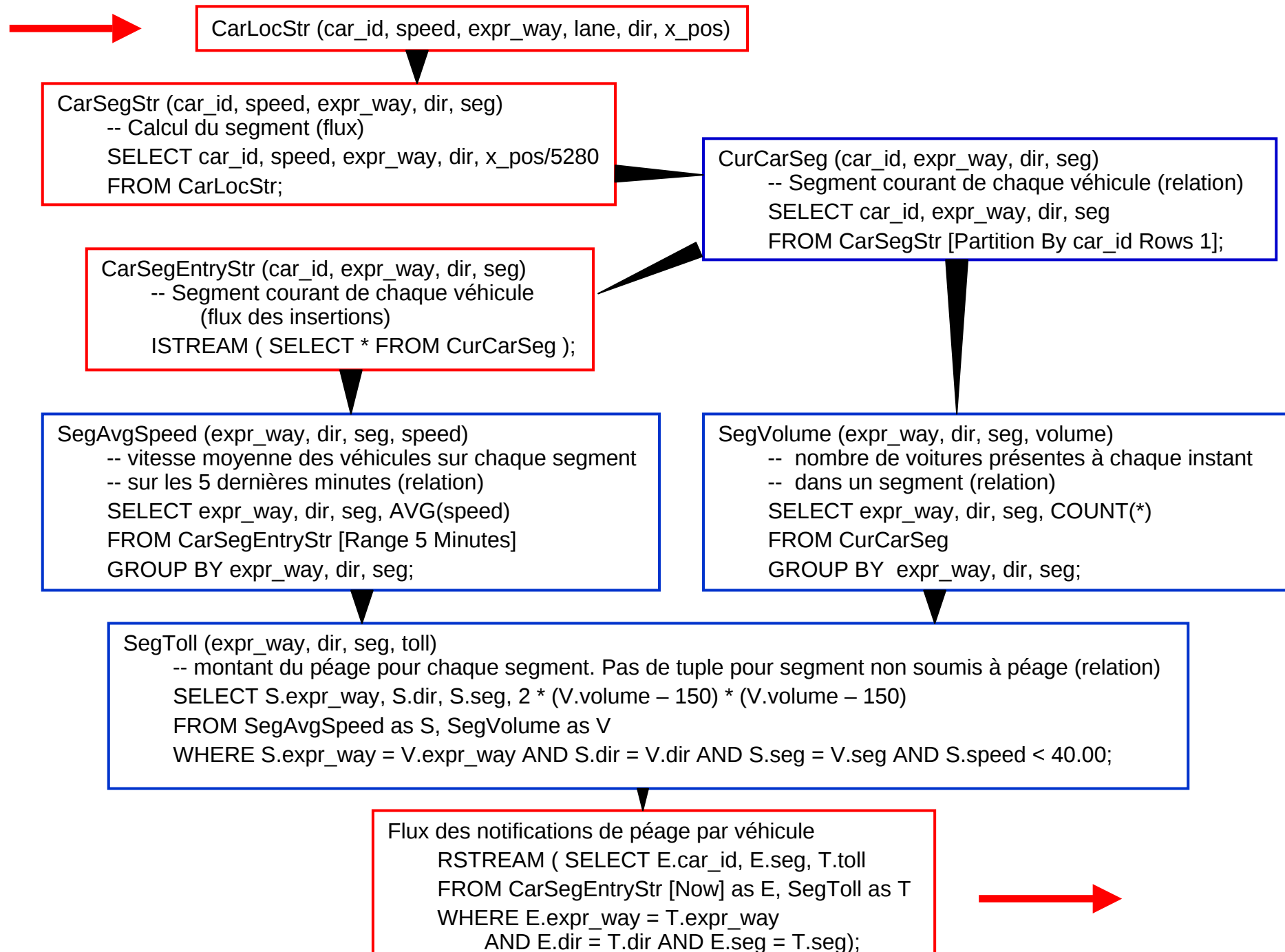
■ Péage dépendant du trafic

- Notification d'un prix à l'entrée d'un nouveau segment / Facturation à la sortie d'un segment
- 5' pour notifier le prix après réception du position report indiquant le changement de segment
- Latest Average Velocity : vitesse moyenne des véhicules d'un segment et d'une direction pendant les 5 minutes précédant la minute du 'position report' déclenchant le calcul
- Toll :
 - gratuit si LAV > 40 MPH ou si NB véhicules dans le segment < 50
 - gratuit si accident détecté dans les 4 segments suivants
 - $2 * (\text{numvehicules} - 50)^2$
- Accident détecté : si au moins 2 véhicules sont arrêtés dans le segment et la voie pendant 4 position reports
- Notification d'accident aux véhicules (peuvent réagir et change de route)

Exemple STREAM « Linear Road » (3)

■ Système STREAM

- Université de Stanford
- DSMS généraliste
- Langage CQL (Continuous Query Language) déclaratif basé sur SQL
- Deux structures :
 - Flux : ordonné par un timestamp entier attribué par le système à l'arrivée des n-uplets
 - Relation : table dont le contenu varie au cours du temps
- Fenêtrage (fenêtres glissantes logiques ou physiques, partitionnables)
- Les clauses de fenêtrage créent des relations
- Opérateurs relation-flux :
 - ISTREAM : flux des insertions
 - DSTREAM : flux des suppressions
 - RSTREAM : flux du contenu de la relation à chaque instant
- Définition d'un ensemble de flux et de requêtes
- Nouveau système développé 'from scratch', génération d'un plan d'exécution optimisé mais **n'accepte pas de nouvelle requête**
- DSMS de démo sur le site de STREAM : <http://www-db.stanford.edu/stream>
- Téléchargeable mais projet arrêté début 2006



Architecture des DSMS (3)

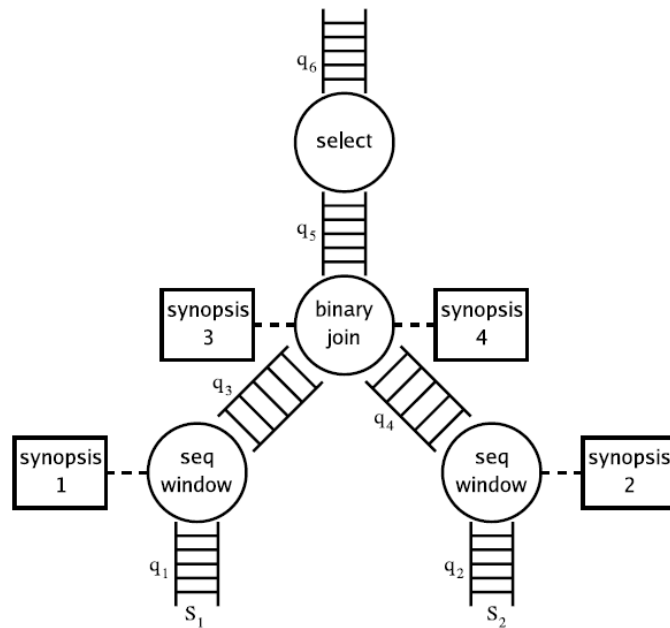


Fig. 2. A simple query plan illustrating operators, queues, and synopses

Source : STREAM (Arasu *et al.* 2004)

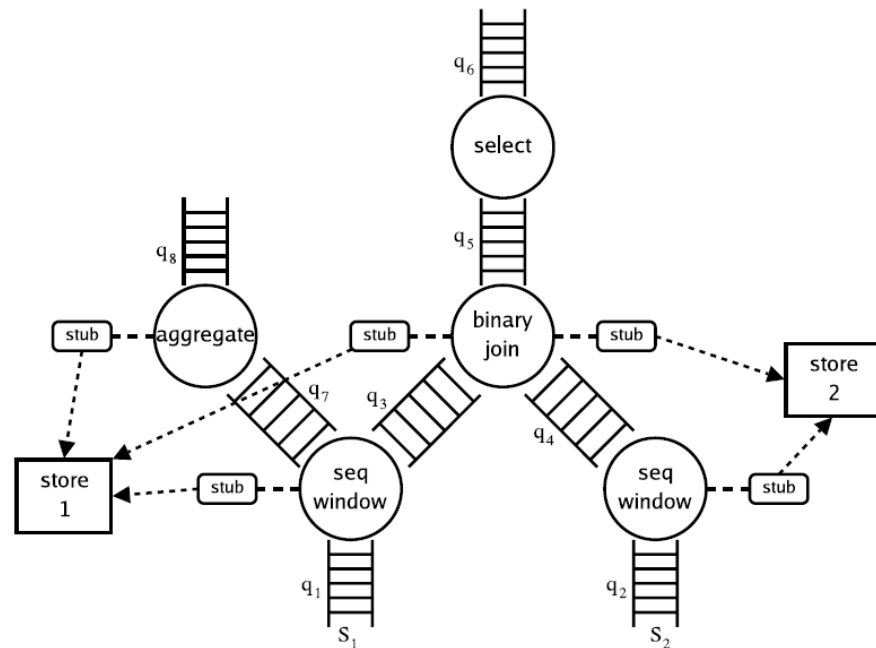
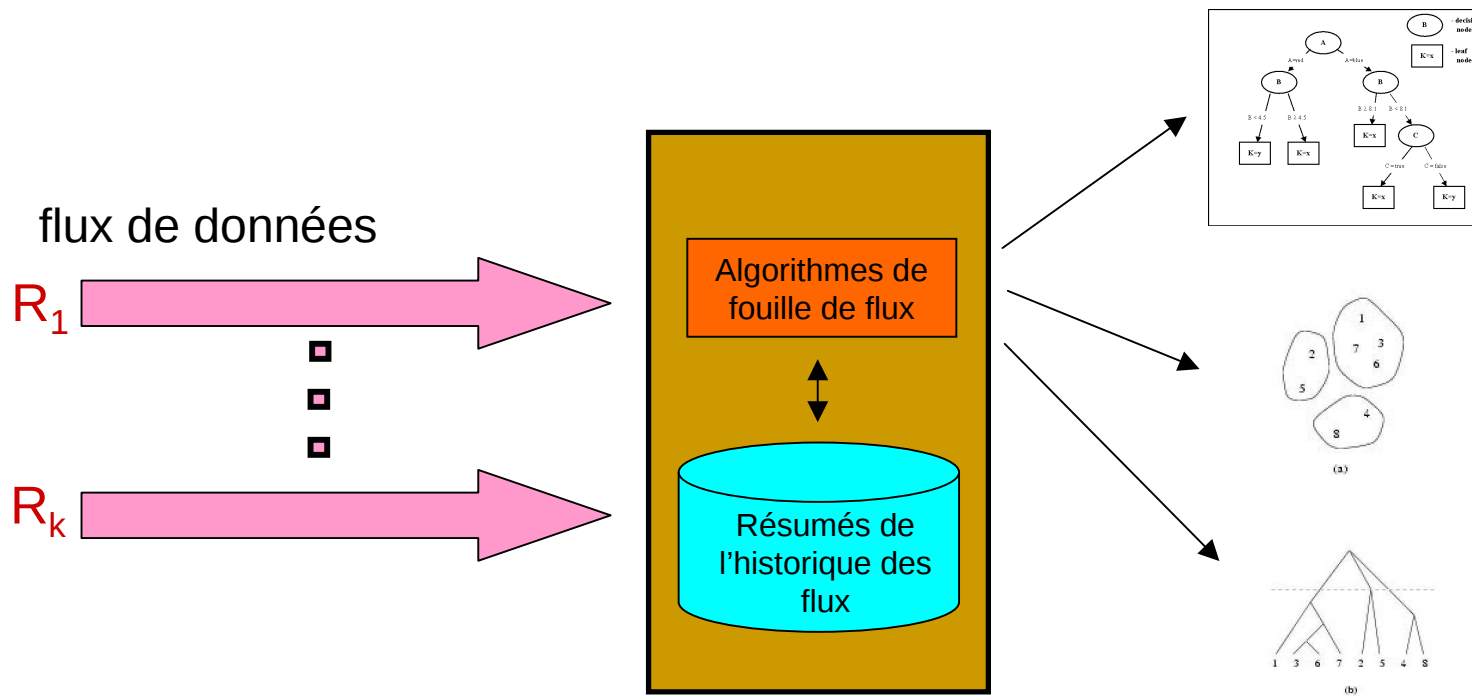


Fig. 3. A query plan illustrating synopsis sharing.

à suivre

■ techniques d'analyse des flux de données



Plan

- Système d'information et flux de données
 - Fabrice CLEROT
- Systèmes de gestion des flux de données
 - Georges HEBRAIL
- Techniques d'analyse de flux de données
 - en un point d'observation, Baptiste CSERNE
 - distribuées, Fabrice CLEROT

Plan

- Objectifs et Définition
 - Définition
 - Contraintes
 - Exemples
 - Problématique
- Approches
 - La Gestion Mémoire
 - La Gestion Temporelle
- Techniques Avancées
 - CluStream
 - StreamSamp
- Conclusion

Définition

- Accomplir les tâches de fouille de données classiques sur un flux de données

- Exemples :
 - ❑ ACP
 - ❑ Classification automatique sur flux
 - ❑ Estimer le profil des données
 - ❑ Estimer des indicateurs
 - ❑ Compter des événements et des effectifs
 - ❑ ...

Contraintes classiques

- Obéir aux contraintes classiques des flux de données :
 - ❑ Ressources mémoire limitées
 - ❑ Ressources CPU limitées
 - ❑ Une seule passe par élément

Exemple : Les Arbres de Décision

■ Objectif

- ❑ Arbres de Décisions Très Rapides (VFDT).
- ❑ Réaliser un arbre de décision à partir de N exemples
- ❑ Attributs discrets
- ❑ On dispose d'une fonction de coupe à maximiser $G(X)$

■ Inspiration

- ❑ Les Arbres de Hoeffding

Exemple : Les Arbres de Hoeffding

(Domingos et. Al. 2000)

- A chaque nouvel exemple ($a_1, a_2, \dots, a_d$).
 - Insérer l'exemple le plus loin possible dans l'arbre.
 - Si tous les exemples à une feuille ne sont pas de la même classe
 - Calculer pour l'ensemble des exemples de chacune de ces feuilles $\Delta G = G(a_i) - G(a_j)$.
 - Si $\Delta G > \epsilon(n)$ la feuille est coupée en utilisant a_i .
- Garanties
 - Avec une probabilité $1-\delta$ le choix de a_i est correct. ($\epsilon(n)$ fonction de δ et du domaine de définition.)

Exemple : Les Arbres de Décision

VFDT (Domingos et. Al. 2000)

- Utilise un Arbre de Hoeffding avec des ajustements
 - ❑ Égalités : Si $\Delta G < \varepsilon(n) < \tau$ la coupe est tout de même effectuée (τ défini par l'utilisateur).
 - ❑ Calculs : G n'est calculé aux feuilles qu'après qu'un nombre N_{min} d'exemples y aient été accumulés.
 - ❑ Mémoire : Les feuilles les moins prometteuses sont désactivées. (critère : minimiser taux d'erreur
* proba qu'un exemple tombe dans cette feuille)

Exemple : K Médianes : Principe

■ Objectif

- ❑ Classification automatique sur attributs continus
- ❑ Trouver k éléments centraux dans le flux, tels que la somme des distances de chaque élément du flux à son centre le plus proche soit minimale.

■ Inspiration

- ❑ K médianes classique / Divide and Conquer

Exemple : K Médianes

Small Space (Ghua et. al. 2000)

- ❑ Initialisation : Effectuer 2k médianes sur les m premiers points.
- ❑ Répéter l'opération jusqu'à avoir traité $m^2/2k$ points pour obtenir m médianes intermédiaires
- ❑ Effectuer 2k médianes sur les m médianes intermédiaires d'ordre 1 pour obtenir 2k médianes d'ordre 2.
- ❑ Répéter l'opération en maintenant au plus m médianes d'un ordre donné.
- ❑ Effectuer k médianes sur l'ensemble des médianes intermédiaires

Problématique :

Le passé domine les analyses

- Les algorithmes ne tiennent pas compte d'évolutions dans la distribution des données du flux. (Concept Drift)
- Les méthodes ne permettent pas de sélectionner à posteriori une partie du flux à analyser.

Approches

- Objectifs et Définition
 - Définition
 - Contraintes
 - Exemples
 - Problématique
- Gestion Mémoire
 - L'Échantillonnage
 - Les Agrégats
 - Les Histogrammes
 - Les Sketchs
- Gestion Temporelle
 - Système de Clichés
- Techniques Avancées
 - CluStream
 - StreamSamp

Gestion de l'Espace Mémoire : L'Échantillonnage

- Difficultés :
 - Échantillonnage sur fenêtres glissantes.
- Deux familles d'algorithmes selon les fenêtres :
 - Pour les fenêtres logiques.
 - Pour les fenêtres physiques.
- Notations
 - Échantillons de taille k
 - Fenêtre de taille n

Échantillonnage non Fenêtré : Échantillonnage Réservoir (Vitter 1985)

■ Principe

- Un réservoir de taille R est maintenu en permanence.
- De ce réservoir peuvent être extraits les k échantillons.

■ Initialisation

- Sélectionner les k premiers éléments.

■ A chaque élément E d'indice $i+1$ pour $i > k$:

- L'élément est sélectionné avec pour probabilité $k/(i+1)$.
- L'élément qu'il remplace est choisi au hasard dans le réservoir.

■ Défaut

- Ne prend pas en compte les suppressions d'éléments.

Échantillonnage sur Fenêtre Logique : Échantillonnage avec Réserve (Babcock et. al. 2000)

- Initialisation
 - Placer les k premiers éléments dans la réserve.
- A chaque élément E d'indice i pour $i > k$:
 - L'élément est sélectionné avec pour probabilité $2.c.k.\log(n) / n$.
 - Si un élément expire il est retiré de la réserve.
- Garantie
 - La taille de la réserve est entre k et $4.c.k.\log(n)$.

Échantillonnage sur Fenêtre Logique : Échantillonnage en Chaîne (Babcock et. al. 2000)

■ Initialisation

- K chaînes à construire
- Premier élément de chaque chaîne choisi avec pour probabilité $1/\text{Min}(i,n)$.

■ Pour chaque chaîne.

- L'élément suivant est choisi au hasard dans l'intervalle d'indices $[i+1, i+n]$ et stocké à son arrivée.
- Un nouvel élément suivant est alors sélectionné.
- Dès qu'un élément expire, il est retiré de la chaîne

■ L'échantillon est formé des têtes des k chaînes.

Gestion de l'Espace Mémoire :

Les Agrégats

- Définis pour un groupe d'éléments A de dimension d et contenant n éléments A_i de type $(t, a1_i, a2_i, \dots, ad_i)$.
- Donne des informations de nature résumée sur le groupe d'éléments
- Exemple
 - Les moments

Les Agrégats : Le Cluster Feature Vector (CFV) (BIRCH, Zhang 1996)

- De structure :

$(n, CF1(a1), CF2(a1), \dots, CF1(ad), CF2(ad))$.

- Avec

- $CF1(ak) = \sum_{i=1}^n (ak_i)$

- $CF2(ak) = \sum_{i=1}^n (ak_i)^2$

- Remarque

- Les CFV supportent les opérations ensemblistes.

Les Agrégats : Le CFV, variante. (Aggarwal 2002)

- De structure :

$(n, CF1(t), CF2(t), CF1(a1), CF2(a1), \dots, CF1(ad), CF2(ad))$.

- Avec

- $CF1(ak) = \sum_{i=1}^n (ak_i)$
- $CF2(ak) = \sum_{i=1}^n (ak_i)^2$

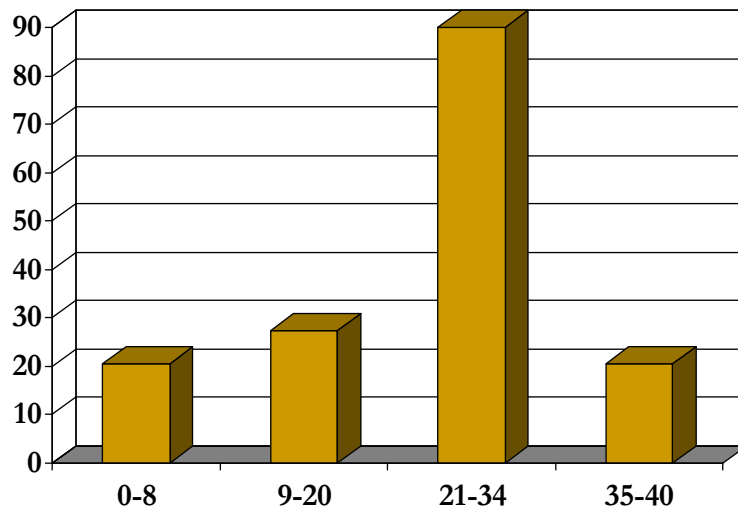
- Remarque

- Le temps occupe maintenant la place d'une variable comme une autre.

Gestion de l'Espace Mémoire :

Les Histogrammes

Histogramme Simple



■ Avantages

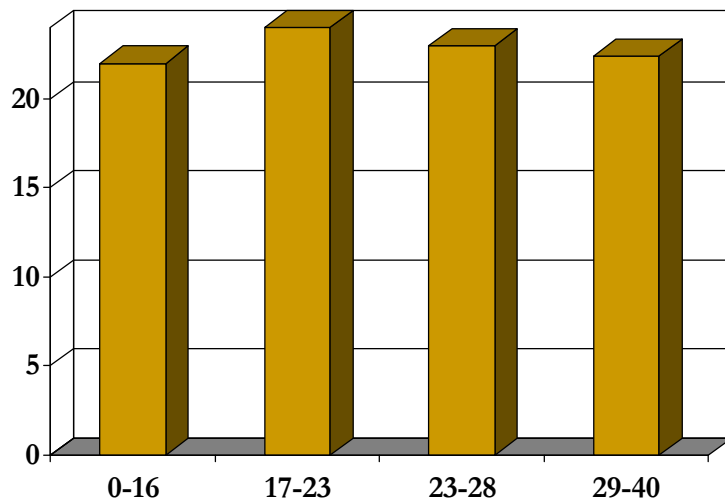
- Simple
- Calcul de Quantiles

■ Défauts

- Demande une connaissance préalable de la distribution

Les Histogrammes : Histogramme Optimal (Jagadish et al. 1998)

Histogramme Optimal



■ Avantages

- Résumé plus d'information
- Calcul de Quantiles
- Pas de connaissances préalables.

■ Défauts

- Coûteux en calculs

Gestion de l'Espace Mémoire :

Les Sketchs

- Un résumé d'un type plus sophistiqué que les agrégats ou les histogrammes.
- Un résumé souvent destiné à une tâche précise, comme par exemple évaluer une norme.
- Exemples :
 - Les projections sur des sous espaces.
 - Aléatoires
 - Ondelettes
 - ACP locales
 - Les comptages particuliers
 -

Les Sketches : COUNT (1/3)

(Flajolet 1985)

■ Objectif :

Compter le nombre n de valeurs distinctes dans un flux.

■ Structure :

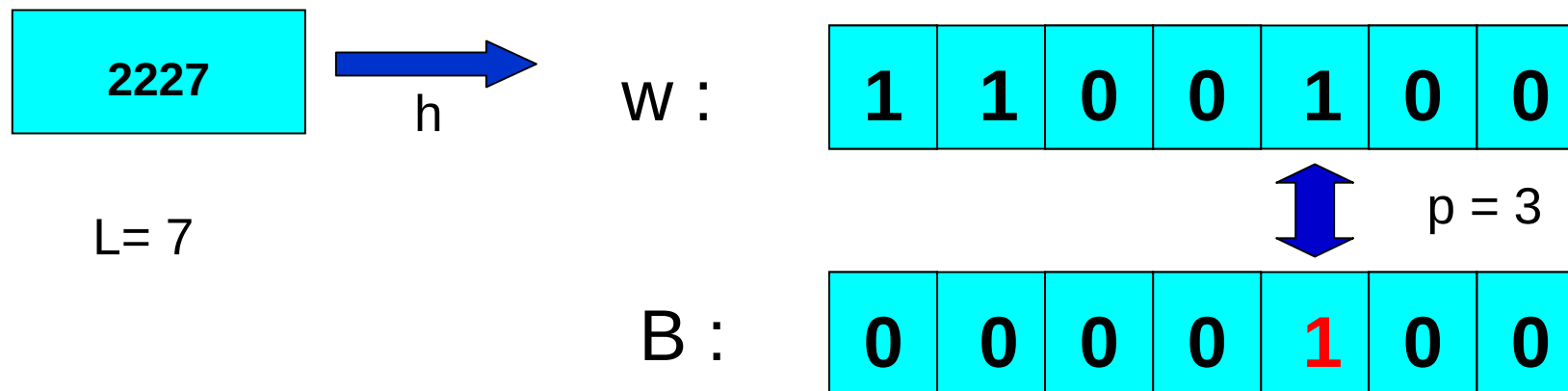
- Un mot B de L bits initialement tous à 0.
- Une fonction de hachage h transformant un élément du flux en un mot binaire de taille L .
- h telle que l'ensemble des valeurs possibles se distribuent uniformément sur l'ensemble des mots binaires de taille L .

Les Sketches : COUNT (2/3)

(Flajolet 1985)

■ Méthode

- Pour chaque nouvel élément X :
 - Réaliser $w = h(X)$
 - Sélectionner la position p du bit à 1 le moins significatif.
 - Mettre le bit à la position p dans B à 1.



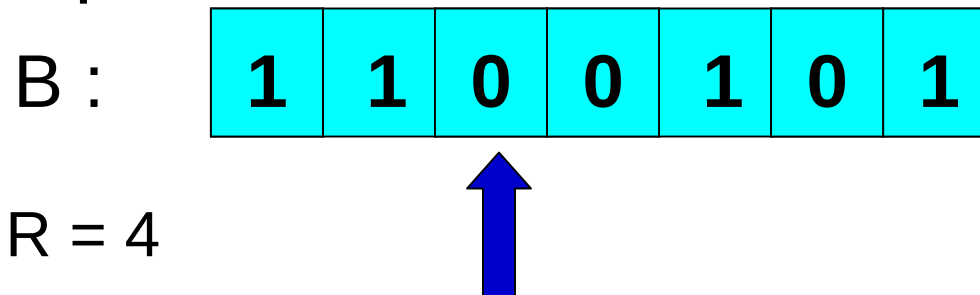
Les Sketches : COUNT (3/3)

(Flajolet 1985)

■ Résultats :

- ❑ Soit R la position du 0 le plus à gauche dans B .
- ❑ $E(R) = \log_2 (\varphi * n)$ avec $\varphi = 0.77351...$
- ❑ $\sigma(R) = 1.12$

■ Exemple



Les Sketches : Count Min (1 / 3)

(Cormode et Muthukrishnan 2003)

■ Problématique

- ❑ Sur un flux de données suivant le modèle du tiroir caisse avec :

le signal : $a_1(t) a_2(t) \dots a_D(t)$ (D grand)

les m^àj : $(i(t), c(t)) \rightarrow a_i(t) = a_i(t-1) + c(t)$
 $a_j(t) = a_j(t-1), i \neq j$

- ❑ Pouvoir répondre aux requêtes de type :

- Donner $Q(i) = a_i(t)$
- Donner $Q(l, r) = \sum_{i=l}^r a_i$
- Donner $Q(a, b) = \sum_{i=a}^b a_i * b_i$

Les Sketches : Count Min (2/3)

(Cormode et Muthukrishnan 2003)

- Construction d'un Sketch CM de param. (w, d)
 - une matrice de taille $w \times d$ peuplée de 0.
 - d fonctions de hachage indépendantes 2 à 2.
 - Pour chaque update (i, c) pour chaque ligne de la matrice, une case est incrémentée de c .

Cette case est déterminée par la fonction de hachage associée à la ligne.

Soit pour tout j :

$\text{count}[j, h_j(i)] \leftarrow \text{count}[j, h_j(i)] + c$

Les Sketches : Count Min (3/3)

(Cormode et Muthukrishnan 2003)

- Réponses

- Donner $Q(i) = a_i(t)$

- $a_i(t) = \min(j) \text{ count } [j, h_j(i)] \text{ avec :}$

- $\hat{a}_i(t) < a_i + (e / w) * ||a||$

- En utilisant plusieurs sketches, permet aussi de répondre à toutes les autres requêtes.
- Permet par déduction de répondre à des requêtes plus complexes

Les Sketches : Count Sketch(1 / 3)

(Charikar et. al. 2004)

■ Objectif :

Pour un flux d'éléments E_i de fréquences N_i tels que N_1 soit l'élément le plus fréquent. Trouver une liste de k éléments

telle que : $N_i < (1-\epsilon)N_k$

■ Structure :

- Une matrice de comptages $t \times b$
- t fonctions de hachages r_i de E vers $\{+1; -1\}$
- t fonctions de hachages h_i de E vers $\{1 \dots b\}$

Les Sketches : Count Sketch(2/3)

(Charikar et. al. 2004)

- Ajouter un nouvel élément E :
 - Pour chaque fonction h_i : $h_i[E] += s_i[E]$
 $1 \dots b$

1...t		+1		
			-1	
	-1			
				+1

- Estimer l'effectif d'un élément E :
 - Calculer la médiane sur i de $\{h_i[E] \cdot s_i[E]\}$

Les Sketches : Count Sketch(3/3)

(Charikar et. al. 2004)

■ Résolution

- ❑ Ajouter chaque nouvel élément E_i
- ❑ Si E_i est déjà dans la liste :
 - Incrémenter son compteur.
- ❑ Sinon :
 - Si son compte estimation est plus grande que le plus petit compte de la liste :
 - ❑ Ajouter E_i à la liste.
 - ❑ Retirer de la liste l'élément au compte le plus faible.

Norme de Hamming

- En se plaçant dans le même modèle, on peut définir la norme de Hamming pour un flux :

$$\|a\|_h = \{ i, a_i \neq 0 \}$$

- Si le vecteur a représente des effectifs, $\|a\|_h$ représente le nombre d'éléments dont le compte est non nul.

Gestion de l'Espace Mémoire :

Les Autres

- Combiner des techniques de résumés avec des techniques de regroupements de points similaires :
 - ❑ Classifications
 - ❑ Grilles
 - ❑ Règles
 - ❑

Gestion Temporelle

■ Principe

Établir une structure de stockage des données permettant de reconstituer le passage du flux au cours d'une période donnée.

■ Structures Temporelles

- ❑ Clichés.
- ❑ Fenêtres Imbriquées.

Système de Clichés

- L'état du système est sauvegardé à intervalles réguliers.
- La structure de données est choisie de façon à permettre des soustractions entre clichés.
- La distribution des clichées est choisie de façon à avoir plus de clichés proches du temps courant.

Systeme de Clichés :

Exemple de distribution 2^o

Ordre	Clichés (par tic d'horloge)	Pas
0	69 67 65	2^1
1	70 66 62	2^2
2	68 60 52	2^3
3	56 40 24	2^4
4	48 16	2^5
5	64 32	2^6

Techniques Avancées de Résumés

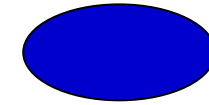
- Objectifs et Définition
 - Définition
 - Contraintes
 - Exemples
 - Problématique
- Gestion Mémoire
 - L'Échantillonnage
 - Les Agrégats
 - Les Sketchs
- Gestion Temporelle
 - Système de Clichés
 - Fenêtres Inclinées
- Techniques Avancées
 - CluStream
 - StreamSamp

CluStream (Aggarwal 2002)

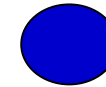
- Algorithme basé sur trois principes :
 - ❑ Division du traitement en deux parties, une en ligne, et une hors ligne.
 - ❑ Création et maintenance de micro clusters.
 - ❑ Archivage régulier de l'état de ces micro clusters dans des clichés.

CluStream (1 / 4) (partie en ligne)

- Initialisation
 - Constitution des Micro Clusters d'origine hors ligne.
- Pour Chaque élément
 - Localisation du Micro Cluster le plus proche.
 - Test d'admission
 - Si admis, MàJ du CFV
 - Sinon, création d'un nouveau Micro Cluster après destruction d'un ancien.



Micro Cluster 1
(CFV, liste d'ID)



Micro Cluster 2
(CFV, liste d'ID)

....



Micro Cluster N
(CFV, liste d'ID)

CluStream (2/4) (partie en ligne)

■ Destruction d'un Micro Cluster

- ❑ Détruire un micro cluster trop vieux.
(critère basé sur la date d'arrivée des derniers éléments)
- ❑ Fusionner les deux Micro Cluster les plus proches.
(MàJ de la liste d'ID des Micro Clusters)

CluStream (3/4) (partie en ligne)

- Archivage

- Système de Clichés avec une distribution 2^o

- Chaque cliché contient

- Le CFV de chaque Micro Cluster
 - La liste d'ID de chaque Micro Cluster

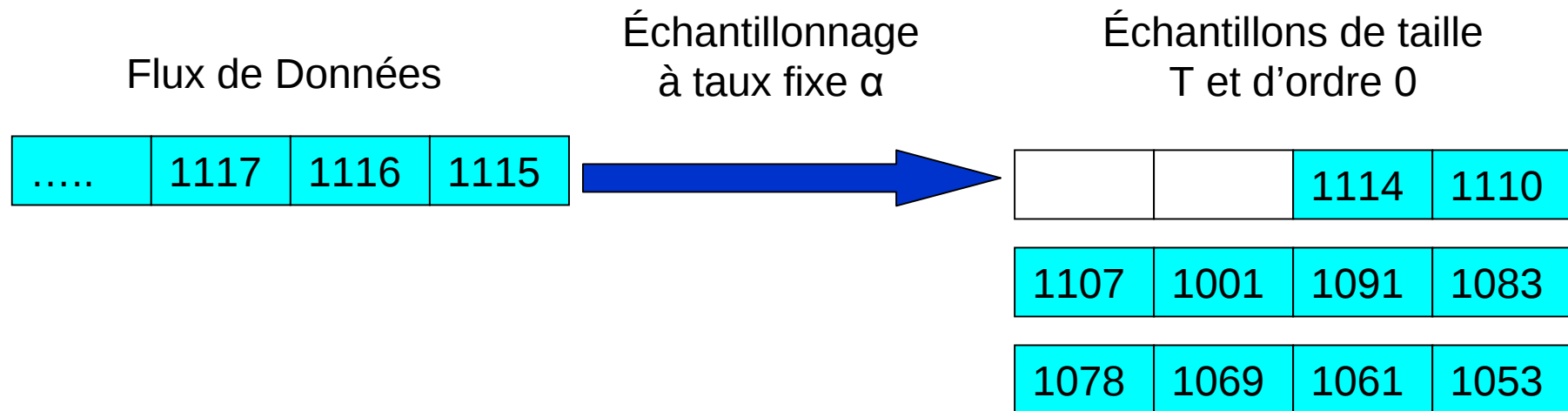
CluStream (4/4) (partie hors ligne)

- Reconstitution à partir des clichés de la partie du flux à analyser.
(sous forme de micro clusters)
- Application d'algorithmes de fouilles de données classique aux micro clusters.
- Le résultat de ces algorithmes est le résultat final.

StreamSamp (Csernel 2006)

- Algorithme basé sur trois principes :
 - L'échantillonnage du flux
 - Le Stockage des échantillons dans une structure de fenêtres inclinées.
 - Séparation d'une partie en ligne et hors ligne

StreamSamp (1/3) (partie en ligne)



Jusqu'à L échantillons d'un
ordre o donné.

StreamSamp (2/3) (partie en ligne)

$L = 3$

Échantillons de taille T et d'ordre o .

			1110
1107	1001	1091	1083
1078	1069	1061	1053
1048	1036	1031	1022

Échantillons de taille T et d'ordre $o + 1$.

1078	1061	1036	1022
1004	983	962	944
918	802	786	764

Tirage aléatoire de $T/2$ éléments de chaque échantillon père

StreamSamp (3/3) (partie hors ligne)

- Les échantillons portant sur la partie du flux à analyser sont fusionnés.
- Chaque élément d'un fichier d'ordre o reçoit une pondération de 2^o
- Application d'algorithmes de fouilles de données classique à cet échantillon représentatif du flux.

Conclusion

- Nombreuses techniques.
- Nombreuses approches différentes.
- A adapter en fonction des résultats souhaités et de la nature des données, comme en fouille de données classique.

Bibliographie (Analyse de Flux de Données)

Mining High-Speed Data Streams. P. Domingos and G. Hulten. In Proceedings of the 6th ACM SIGKDD conference, Boston, USA, pages 71-80, 2000.

Clustering data streams. S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan. In IEEE Symposium on Foundations of Computer Science. IEEE, 2000.

Random sampling with a reservoir. J. Vitter. ACM Trans. Math. Softw., 11(1) :37–57, 1985.

Sampling from a moving window over streaming data. B. Babcock, M. Datar, and R. Motwani. In Proceedings of the thirteenth annual ACM-SIAM.

Birch : an efficient data clustering method for very large databases. T. Zhang, R. Ramakrishnan, and M. Livny. In Proceedings of the SIGMOD 1996 Conference, pages 103–114, 1996.

Optimal histograms with quality guarantees. H. Jagadish, N. Koudas, S. Muthukrishnan, V. Poosala, K. Sevcik, and T. Suel. In *Proc. of the 1998 Intl. Conf. on Very Large Data Bases*, pages 275–286, 1998.

Probabilistic Counting Algorithms for Data Base Applications. P. Flajolet. In Journal of Computer and System Sciences, Volume 32, Issue 2, page 182-209, Septembre 1985

Finding frequent items in data streams. M. Charikar, K. Chen, and M. Farach-Colton. Theor. Comput. Sci., 312(1) :3–15, 2004.

A framework for clustering evolving data streams. C.C. Aggarwal, J. Han, J. Wang, and P.S. Yu. In Proceedings of the 29th VLDB Conference, Berlin, Germany, 2003.

StreamSamp DataStream Clustering Over Tilted Windows through Sampling. B. Csernel, F. Clerot, G. Hébrail. In Knowledge Discovery from Data Streams Workshop (ECML/PKDD), Berlin, September 2006.

Plan

- Système d'information et flux de données
 - Fabrice CLEROT
- Systèmes de gestion des flux de données
 - Georges HEBRAIL
- Techniques d'analyse de flux de données
 - en un point d'observation, Baptiste CSERNEL
 - distribuées, Fabrice CLEROT

traitement distribué de flux de données

- un exemple simple illustrant la problématique du traitement distribué:
 - ensemble de collecteurs observant chacun un flux d'identifiants
 - chaque identifiant peut apparaître plusieurs fois et sur différents collecteurs
 - un contrôleur qui doit connaître le nombre d'identifiants différents qui ont été observés par l'ensemble des collecteurs

traitement distribué de flux de données

- exemples d'application:
 - supervision de réseau IP
 - $Id = (@S, @D, pS, pD, Protocole)$
 - collecteur = routeur
 - architecture de communication "sûre"
 - combien de flots différents dans le réseau ?
 - supervision de bâtiment
 - des personnes portant des badges RFID dans un bâtiment
 - des capteurs dans le bâtiment
 - architecture de communication ad hoc, peu sûre
 - combien de gens dans le bâtiment ?

traitement local

- économie de ressources locales
 - un collecteur ne peut pas conserver une liste de tous les Id qu'il a observé
 - chaque collecteur conserve un sketch
 - sketch α -stable avec $\alpha \rightarrow 0$
 - sketch de Flajolet-Martin
- sketch "**combinables**"

- sketch α -stable:
$$\sum_c Sk_c = Sk_{\cup c}$$

- sketch FM:
$$\bigvee_c FM_c = FM_{\cup c}$$

stratégie de collection

- chaque collecteur maintient un sketch
 - mêmes paramètres pour les collecteurs
- les collecteurs transmettent leur sketch au contrôleur
 - sketch α -stable: transmission de la variation du sketch depuis la dernière transmission
 - sketch FM: transmission du sketch
- le contrôleur maintient un sketch global à partir des sketches transmis par les collecteurs

stratégie de collection: borne sur l'erreur

- erreur sur le contrôleur:
 - erreurs associées aux sketches des collecteurs
 - bornées en fonction des ressources disponibles sur les collecteurs
 - erreurs associées au délai entre les transmissions des collecteurs vers le contrôleur
 - chaque collecteur transmet quand il mesure (localement) que sa contribution à l'erreur dépasse un certain seuil
 - seuil absolu: le collecteur peut décider seul
 - seuil relatif: le contrôleur doit diffuser le nombre d'individus différents vus

influence de l'infrastructure de communication

- réseau dédié N-1, collecteurs vers contrôleur
 - pertes négligeables
 - accumulation uniquement au niveau du collecteur
 - les deux types de sketches sont utilisables:
 - les performances du sketch α -stable le rendent préférable
- ad hoc entre collecteurs vers le contrôleur:
 - pertes non négligeables
 - transmissions multiples nécessaires et accumulation dans le réseau ad hoc
 - seul le sketch de Flajolet-Martin est utilisable

$$\text{■ } FM_1 \cup FM_1 \cup FM_2 = FM_1 \cup FM_2$$

traitement distribué de flux de données

- l'erreur sur le contrôleur peut être bornée en contrôlant indépendamment
 - l'erreur "intrinsèque" (sketch local)
 - l'erreur due à la latence de transmission
- le choix de la méthode dépend de façon cruciale des caractéristiques de l'infrastructure de communication

Bibliographie

- G. Cormode, S. Muthukrishnan, and W. Zhuang, *What's different: Distributed, continuous monitoring of duplicate-resilient aggregates on data streams*, *International Conference on Data Engineering* 2006
- G. Cormode and M. Garofalakis, *Sketching streams through the net: Distributed approximate query tracking*, *International Conference on Very Large Data Bases* 2005

MERCI !

