Introduction to data stream querying and mining

Georges HEBRAIL

Workshop Franco-Brasileiro sobre Mineração de Dados

Recife, May 5-7, 2009

Preliminaries

Massive Scale of Data

MassDAL
Massive Data Analysis Lab

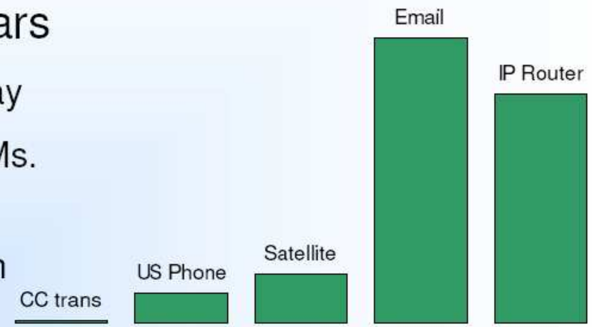
Explosion of Data In Recent Years

- 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.~~ Now at Google

TELECOM ParisTech

Outline

- What is a data stream ?
- Applications of data stream management
- Models for data streams
- Data stream management systems
- Data stream mining
- Synopses structures
- Conclusion

What is a data stream ?

- **Golab & Oszu (2003):** *"A* ***data stream*** *is a* ***real-time****, continuous, ordered (implicitly by arrival time or explicitly by timestamp)* ***sequence of items****. It is impossible to control the order in which items arrive, nor is it feasible to locally* ***store*** *a stream in its entirety."*
- **Structured records ≠ audio or video data**
- **Massive volumes of data, records arrive at a high rate**

Timestamp	Pow. A (kW)	Pow. 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
…	…	…	…	…

What is a data stream ?

- **Golab & Oszu (2003):** *"A **data stream** is a **real-time**, continuous, ordered (implicitly by arrival time or explicitly by timestamp) **sequence of items**. It is impossible to control the order in which items arrive, nor is it feasible to locally **store** a stream in its entirety."*
- **Structured records ≠ audio or video data**
- **Massive volumes of data, records arrive at a high rate**

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
…	…	…	…	…	…

Outline

- What is a data stream ?
- Applications of data stream processing
- Models for data streams
- Data stream management systems
- Data stream mining
- Synopses structures
- Conclusion

TELECOM ParisTech

Applications of data stream processing

Data stream processing

- Process queries (compute statistics, activate alarms)
- Apply data mining algorithms

- **Requirements**
  - Real-time processing
  - One-pass processing
  - Bounded storage (no complete storage of streams)
  - Possibly consider several streams

Applications of data stream processing

Applications

- **Real-time monitoring/supervision of IS (Information Systems) generating unstorable large amounts of data**
  - Computer network management
  - Telecommunication calls analysis (BI)
  - Internet applications (ebay, google, recommendation systems, click stream analysis)
  - Monitoring of power plants

- **Generic software for applications where basic data is streaming data**
  - Finance (fraud detection, stock market information)
  - Sensor networks (environment, road traffic, weather forecast, electric power consumption)

Applications of data stream processing

Let's go deeper into some examples

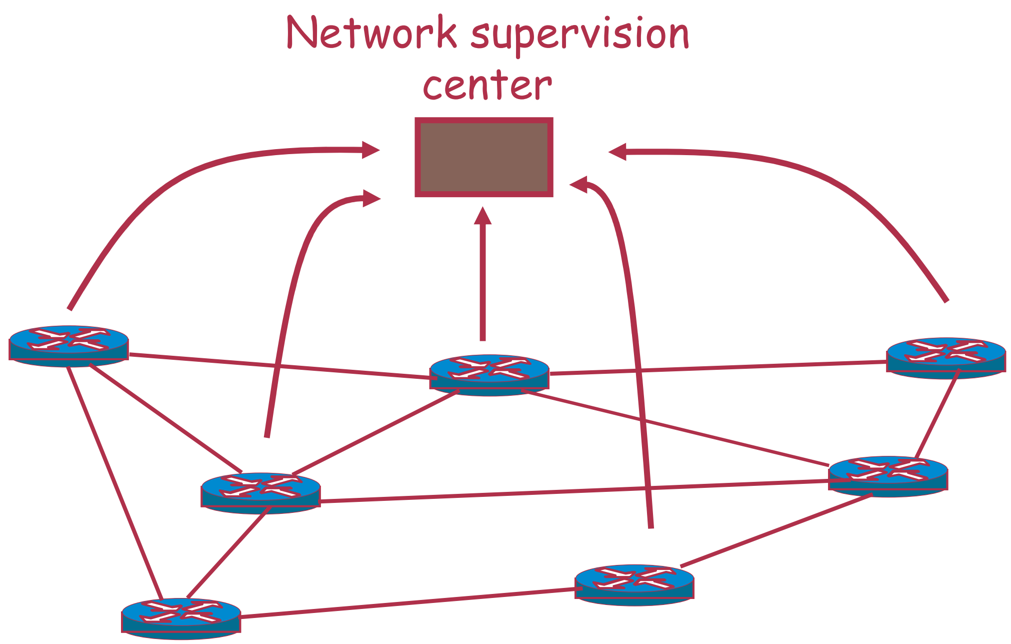
- Network management
- Stock monitoring
- Linear road benchmark

TELECOM ParisTech

Applications of data stream processing

Network management

- Supervision of a computer network
- Improvement of network configuration (hardware, software, architecture)
- Detection of attacks
- Measurements made on routers (Cisco Netflow)



Applications of data stream processing

Network management

- Information about IP sessions going through a router
- Huge amounts of data (300 Go/day, 75000 records/second when sampling 1/100)
- Typical queries:
  - 100 most frequent (@S, @D) on router R1 …
  - How many different (@S, @D) seen on R1 but not R2 …
  - … during last month, last week, last day, last hour ?

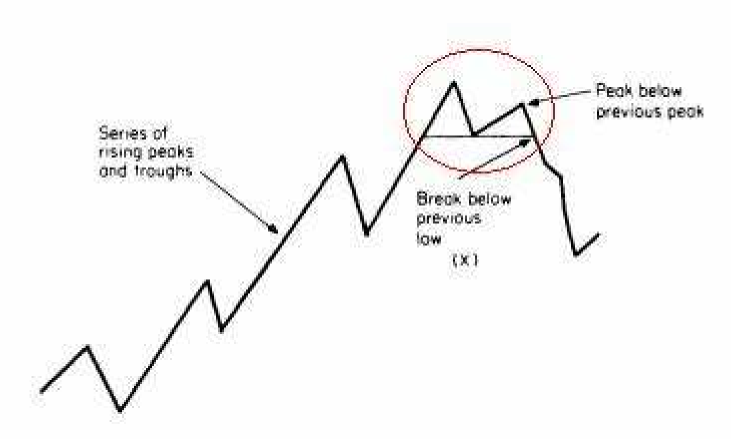
Source	Destination	Duration	Bytes	Protocol
…	…	…	…	…
10.1.0.2	16.2.3.7	12	20K	http
18.6.7.1	12.4.0.3	16	24K	http
12.4.3.8	14.8.7.4	26	58K	http
19.7.1.2	16.5.5.8	18	80K	ftp
…	…	…	…	…

Applications of data stream processing

Stock monitoring

- Stream of price and sales volume of stocks over time
- Technical analysis/charting for stock investors
- Support trading decisions

- ***Notify me when the price of IBM is above $83, and the first MSFT price afterwards is below $27.***
- ***Notify me when some stock goes up by at least 5% from one transaction to the next.***
- ***Notify me when the price of any stock increases monotonically for ≥30 min.***
- ***Notify me whenever there is double top formation in the price chart of any stock***
- ***Notify me when the difference between the current price of a stock and its 10 day moving average is greater than some threshold value***



Source: Gehrke 07 and Cayuga application scenarios (Cornell University)

TELECOM ParisTech

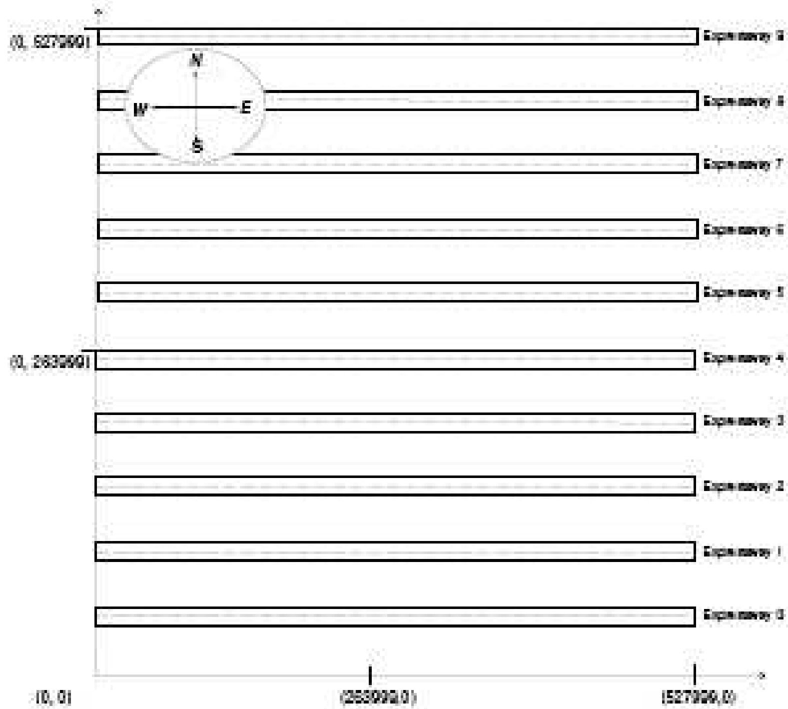
Applications of data stream processing

Linear Road Benchmark

Benchmark to compare Data Stream Management Systems

Linear City

- Imaginary city: 100 miles x 100 miles
- 10 parallel express ways: 2 x (3 lanes + access ramp), cut into segments
- Vehicules send their position every 30’
- Unique clock, no delay on data transmission
- Random generator of vehicule traffic, one accident every 20 minutes

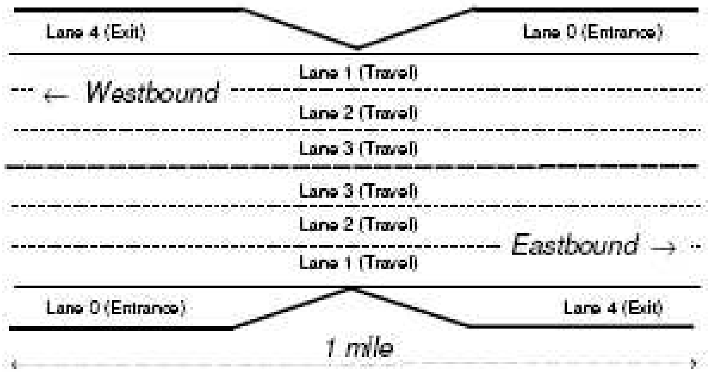


Source: Linear Road: A Stream Data Management Benchmark, VLDB 2004

Applications of data stream processing

Linear Road Benchmark

- Position reports (Time, VID, Spd, Xway, Lane, Dir, Pos)



- Real-time computation of toll

Source: Linear Road: A Stream Data Management Benchmark, VLDB 2004

Applications of data stream processing

Toll depending on traffic

- Notification of a price when entering a new segment, billing when leaving a segment
- Notification within 5' after reception of position reports corresponding to a segment change
- Latest Average Velocity (LAV): average speed of vehicules in a segment and a direction for the last 5 minutes
- Toll :
  - Free if LAV > 40 MPH or if less than 50 vehicules in the segment
  - Free if detected accident in the next 4 segments
  - $2 * (\text{numvehicules} - 50)^2$
- An accident is detected if at least 2 vehicules are stopped in the segment and lane for 4 position reports
- Accidents are notified to vehicules (they can react and change their route)

Source: Linear Road: A Stream Data Management Benchmark, VLDB 2004

TELECOM ParisTech

Outline

- What is a data stream ?
- Applications of data stream processing
- Models for data streams
- Data stream management systems
- Data stream mining
- Synopses structures
- Conclusion

Models for data streams

Structure of a stream

- Infinite sequence of items (elements)
- One item: structured information, i.e. tuple or object
- Same structure for all items in a stream
- Timestamping
  - « explicit »(date field in data)
  - « implicit » (timestamp given when items arrive)
- Representation of time
  - « physical » (date)
  - « logical » (integer)

Models for data streams

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**
...	...	...	...	...	...

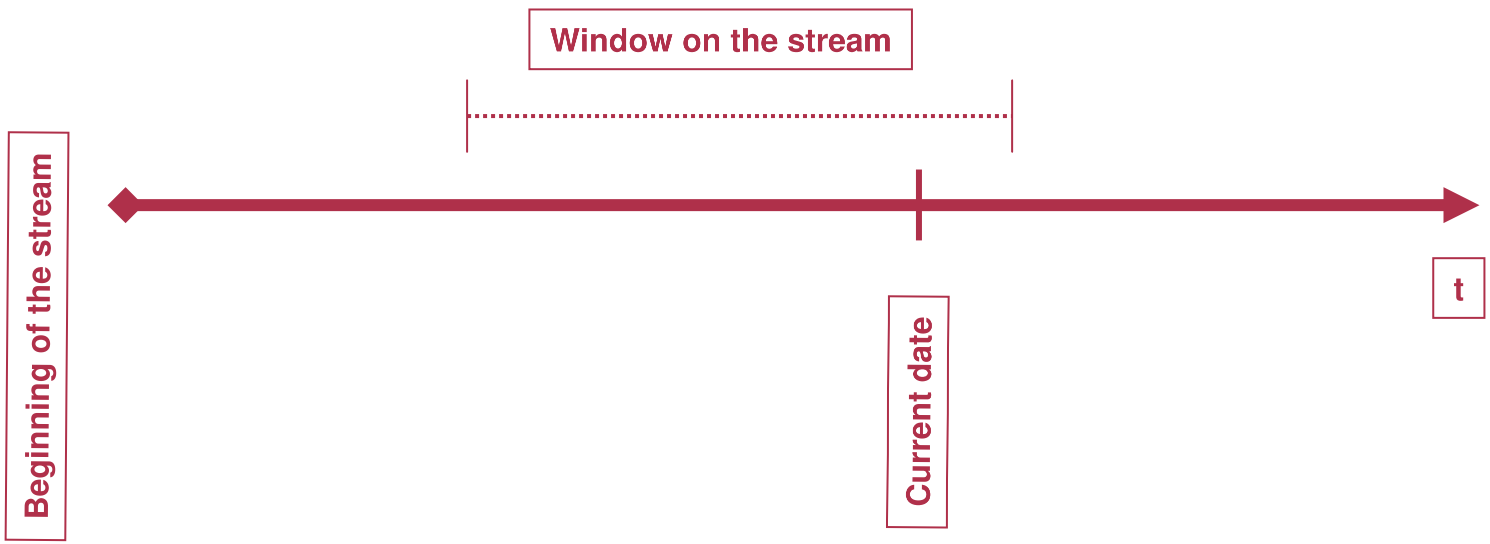
Timestamp	Puis. A (kW)	Puis. R (kVAR)	U 1 (V)	I 1 (A)
...	...	...	...	...
16/12/2006-17:26	**5,374**	**0,498**	**233,29**	**23**
16/12/2006-17:27	**5,388**	**0,502**	**233,74**	**23**
16/12/2006-17:28	**3,666**	**0,528**	**235,68**	**15,8**
16/12/2006-17:29	**3,52**	**0,522**	**235,02**	**15**
...	...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

Models for data streams

Windowing

Applying queries/mining tasks to the whole stream
(from beginning to current time)

Applying queries/mining to a portion of the stream



Models for data streams

Windowing

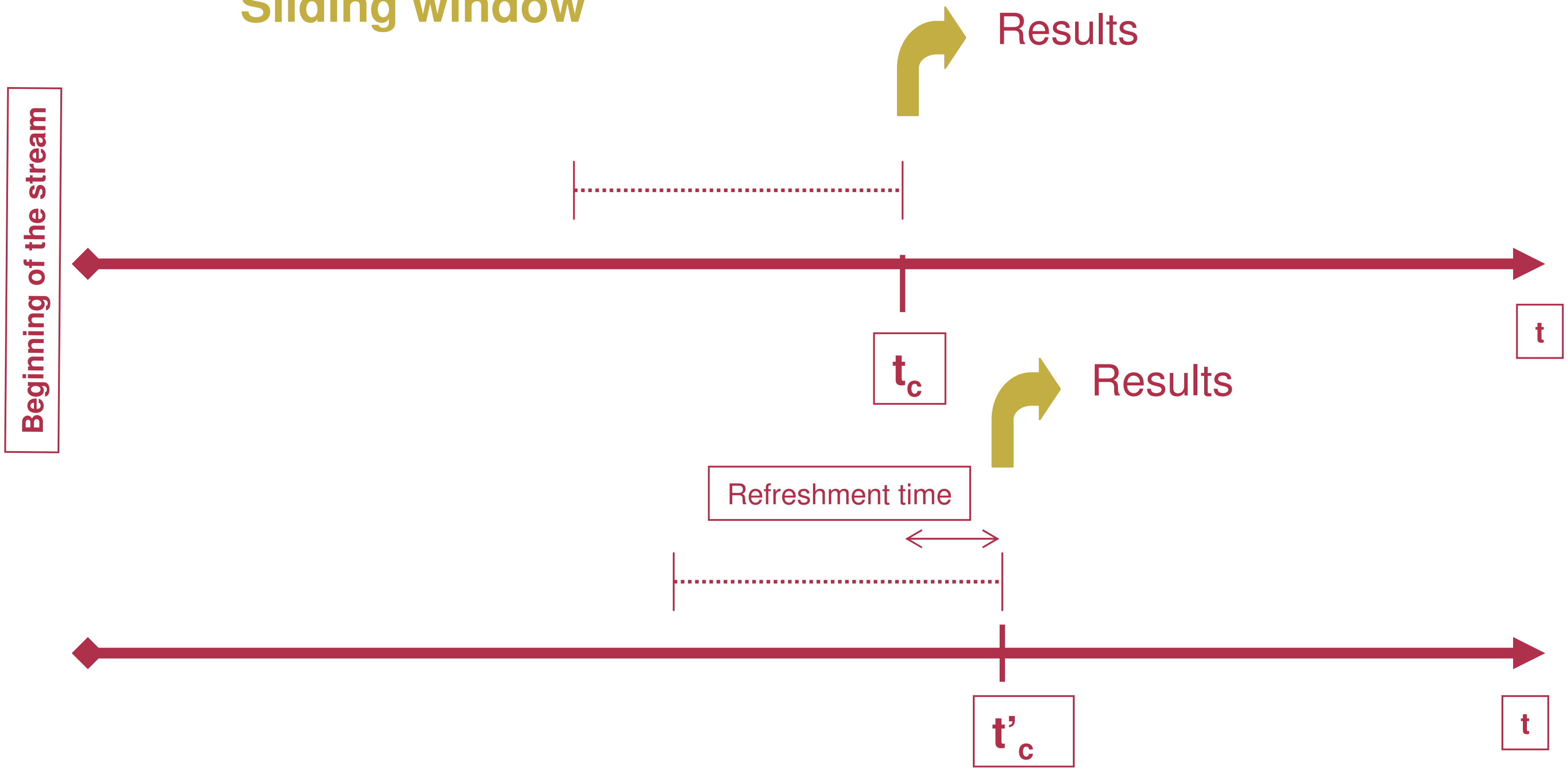
Definition of windows of interest on streams
- Fixed windows: September 2007
- Sliding windows: last 3 hours
- Landmark windows: from September 1st, 2007

Window specification
- Physical time: last 3 hours
- Logical time: last 1000 items

Refreshing rate
- Rate of results production (every item, every 10 items, every minute, …)

Models for data streams

Sliding window



Outline

- What is a data stream ?
- Applications of data stream processing
- Models for data streams
- Data stream management systems
- Data stream mining
- Synopses structures
- Conclusion

DSMS outline

- Definition of a DSMS (Data Stream Management System)
- DSMS data model
- Queries in a DSMS
- Approximate answers to queries
- Main existing DSMS

TELECOM ParisTech

Definition of a DSMS

	DBMS - *Data **Base** Management System*	DSMS - *Data **Stream** Management System*
Data model	Permanent updatable relations	Streams and permanent updatable relations
Storage	Data is stored on disk	Permanent relations are stored on disk Streams are processed on the fly
Query	SQL language Creating structures Inserting/updating/deleting data Retrieving data (one-time query)	SQL-like query language Standard SQL on permanent relations Extended SQL on streams with windowing Continuous queries
Data feeding	SQL language in a programming language Import/export utilities	Tools for capturing input streams and producing output streams (adapters)
Performance	Large volumes of data	Optimization of computer resources to deal with Several streams Several queries Ability to face variations in arrival rates without crash

DSMS outline

- Definition of a DSMS (Data Stream Management System)
- DSMS data model
- Queries in a DSMS
- Approximate answers to queries
- Main existing DSMS

DSMS data model

- **Permanent relations (table)**
  - Tuple (row)
  - Attribute (column)

CUSTOMER TABLE

ID_CUSTOMER	NAME	FIRST	ADRESS	CITY
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

- **Streams**
  - Tuple (row), Attribute (column), Stream of tuples

TIMESTAMP	ID_CUSTOMER	Puis. A (kW)	Puis. R (kVAR)	U 1 (V)	I 1 (A)
…	…	…	…	…	…
16/12/2006-17:26	2	5,374	0,498	233,29	23
16/12/2006-17:27	2	5,388	0,502	233,74	23
16/12/2006-17:26	3	3,666	0,528	235,68	15,8
16/12/2006-17:29	3	3,52	0,522	235,02	15
…	…	…	…	…	…

DSMS data model

- **DSMS output**
  - Updates on permanent tables, for instance:
    - Hourly electric power consumption, aggregated by city, for the last 24 hours
  - One or several output streams, for instance:
    - Alarms to customers with an abnormal consumption during the last 24 hours

DSMS outline

- Definition of a DSMS (Data Stream Management System)
- DSMS data model
- Queries in a DSMS
- Approximate answers to queries
- Main existing DSMS

TELECOM ParisTech

Queries in a DSMS

- Concept of **continuous** queries
  - Standard query in a DBMS: *one-time query*
    - Data are persistent and queries are transient
  - Queries in a DSMS: one-time and continuous queries
    - Standard queries on standard tables
    - Continuous queries when a stream is involved:
      - Executed continuously: permanent queries, transient data
      - Result: output streams or updates on permanent tables
    - Incremental computation of queries (no storage of the whole streams)

Queries in a DSMS: STREAM

STREAM project

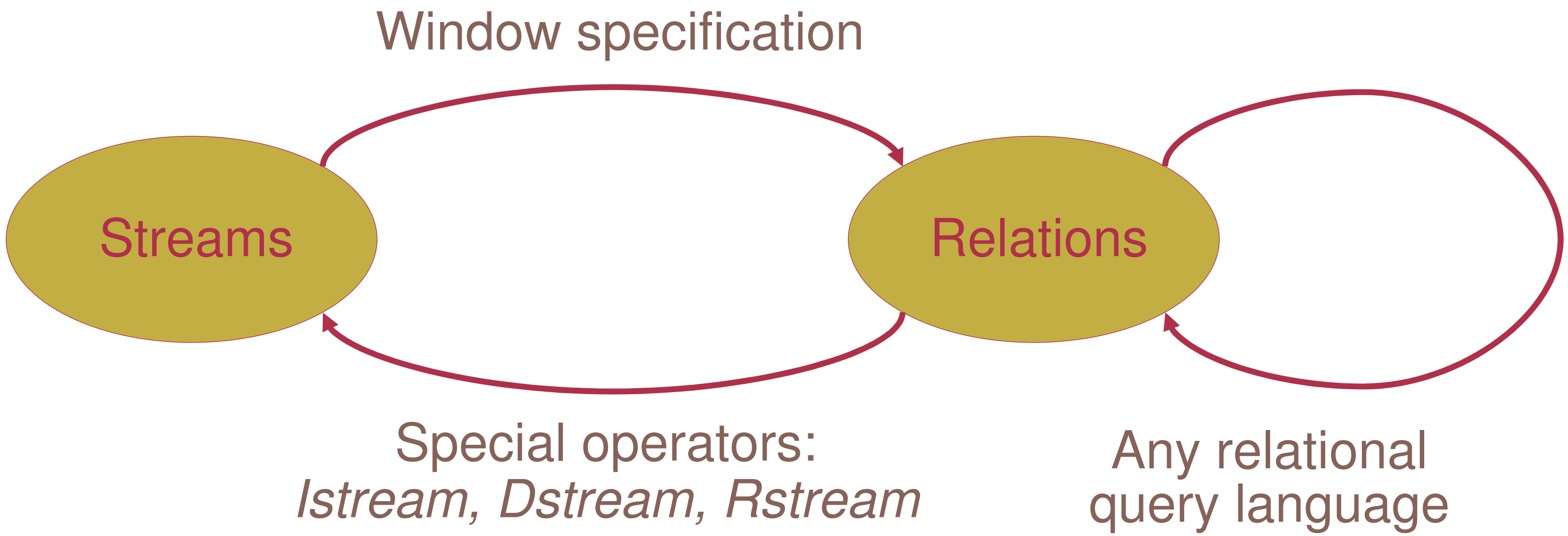
- Stanford University
- General purpose DSMS
- Two structures:
  - STREAMS: implicit logical timestamp
  - RELATIONS : tables with contents varying with time
- CQL Language (Continuous Query Language) based on SQL
- Specification of sliding windows (physical, logical, partitioned)

- Demo site: http://www-db.stanford.edu/stream
- Project ended January 2006

TELECOM ParisTech

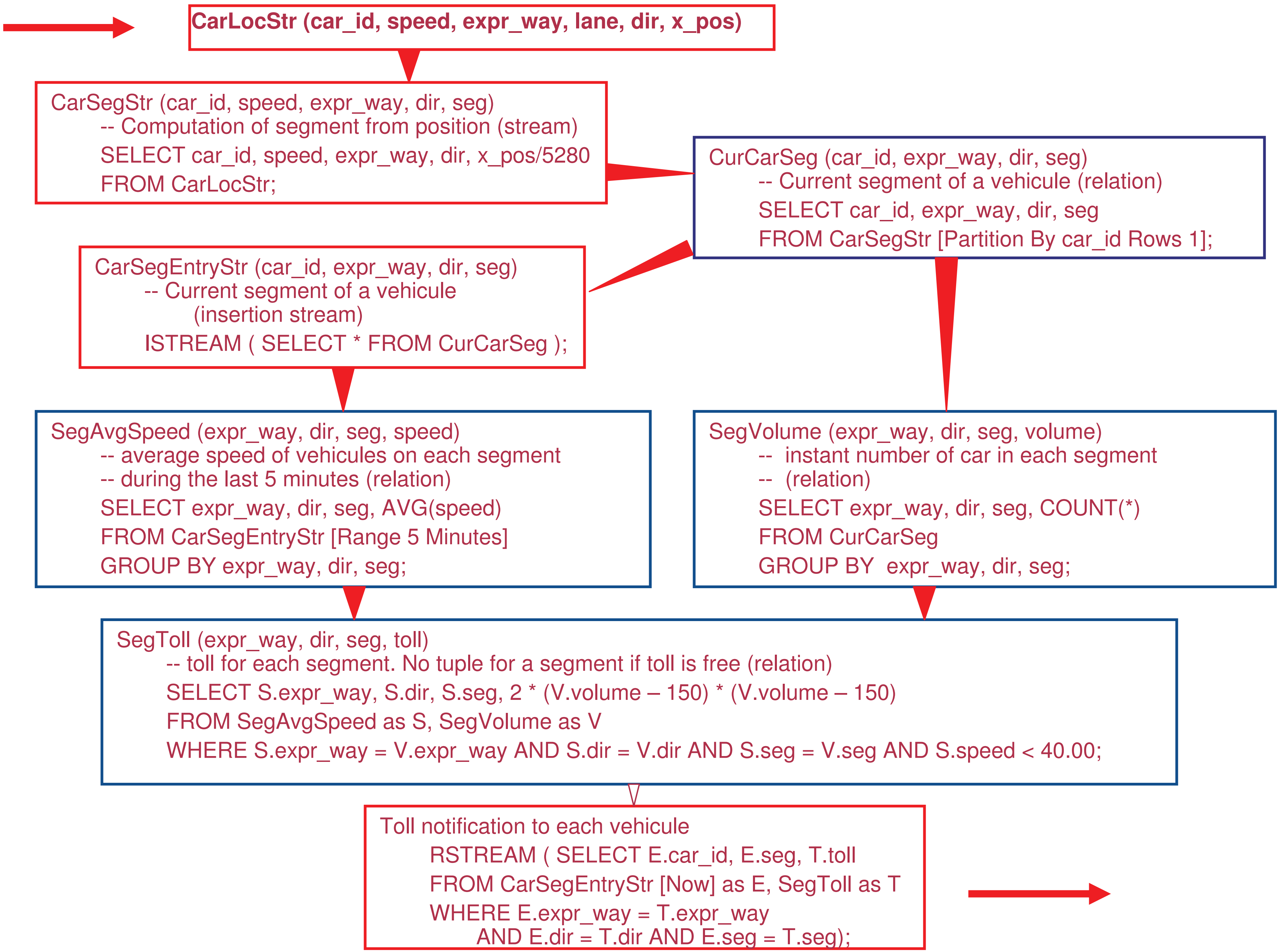
DSMS: STREAM

STREAM – RELATION operators



ISTREAM: stream of inserted tuples
DSTREAM: stream of deleted tuples
RSTREAM: stream of all tuples at every instant

Source: Talk from Jennifer Widom http://infolab.stanford.edu/stream/index.html#talks


CarLocStr (car_id, speed, expr_way, lane, dir, x_pos)
CarSegStr (car_id, speed, expr_way, dir, seg)
-- Computation of segment from position (stream)
SELECT car_id, speed, expr_way, dir, x_pos/5280
FROM CarLocStr;
CurCarSeg (car_id, expr_way, dir, seg)
-- Current segment of a vehicule (relation)
SELECT car_id, expr_way, dir, seg
FROM CarSegStr [Partition By car_id Rows 1];
CarSegEntryStr (car_id, expr_way, dir, seg)
-- Current segment of a vehicule
(insertion stream)
ISTREAM (SELECT * FROM CurCarSeg);
SegAvgSpeed (expr_way, dir, seg, speed)
-- average speed of vehicules on each segment
-- during the last 5 minutes (relation)
SELECT expr_way, dir, seg, AVG(speed)
FROM CarSegEntryStr [Range 5 Minutes]
GROUP BY expr_way, dir, seg;
SegVolume (expr_way, dir, seg, volume)
-- instant number of car in each segment
-- (relation)
SELECT expr_way, dir, seg, COUNT(*)
FROM CurCarSeg
GROUP BY expr_way, dir, seg;
SegToll (expr_way, dir, seg, toll)
-- toll for each segment. No tuple for a segment if toll is free (relation)
SELECT S.expr_way, S.dir, S.seg, 2 * (V.volume – 150) * (V.volume – 150)
FROM SegAvgSpeed as S, SegVolume as V
WHERE S.expr_way = V.expr_way AND S.dir = V.dir AND S.seg = V.seg AND S.speed < 40.00;
Toll notification to each vehicule
RSTREAM (SELECT E.car_id, E.seg, T.toll
FROM CarSegEntryStr [Now] as E, SegToll as T
WHERE E.expr_way = T.expr_way
AND E.dir = T.dir AND E.seg = T.seg);

DSMS outline

- Definition of a DSMS (Data Stream Management System)
- DSMS data model
- Queries in a DSMS
- Approximate answers to queries
- Main existing DSMS

Approximate answers to queries

DSMS challenges

- Generation of execution plans for queries
  - Combination of operators applied to streams + queuing files + temporary storage + scheduler
  - Optimization of use of memory and CPU:
    - Sharing of execution plans, queuing files, buffers, temporary storage
    - Index of queries
  - Dynamic change of execution plans (variations in streams, new queries)
- Quality of service
  - Maintain service in case of scratch, recovery from scratch
  - Maintain service when arrival rates increase

    → Approximate answers to queries

Approximate answers to queries

When ?

- Queries needing unbounded memory
  - Ex : *10 most present IP addresses on a router*
- Too much queries/too rapid streams/too high response time requirements
  - CPU limit
  - Memory limit

Solution: approximate answers to queries

- Sliding windows
- Refreshment rate (*batch processing*)
- Sampling
- Definition of synopses

DSMS outline

- Definition of a DSMS (Data Stream Management System)
- DSMS data model
- Queries in a DSMS
- Approximate answers to queries
- Main existing DSMS

TELECOM ParisTech

Main existing DSMS

General-purpose *research* DSMS's

- STREAM : Stanford University
  - CQL language
  - Query optimization with good memory management
  - Approximate answer with synopses management

- TelegraphCQ : Université de Berkeley
  - Extension of PostgreSQL
  - Continuous queries of CQL type
  - New queries can be added dynamically

- Aurora (Medusa, Borealis) : Brandeis, Brown University, MIT
  - Combination of operators (data flow diagram)
  - Load shedding with explicit definition of quality of service
  - Medusa and Borealis for distributed architecture

Main existing DSMS

Specialized *research or proprietary* **DSMS's**

- Gigascope and Hancock : AT&T
  - Network monitoring
  - Analysis of telecommunication calls
- NiagaraCQ : University of Wisconsin-Madison
  - Large number of continuous queries on web content (XML-QL)
- Tradebot (finance)
- Statstream (statistics)

Commercial DSMS's

- Streambase (cf. Aurora)
- Coral8 (cf. Stream)
- Truviso (cf. TelegraphCQ)
- Aleri
- Esper (open source)

Outline

- What is a data stream ?
- Applications of data stream processing
- Models for data streams
- Data stream management systems
- Data stream mining
- Synopses structures
- Conclusion

Data stream mining outline

- Definition
- Decision tree
- PCA
- Clustream

Data stream mining: definition

Goal

Apply data mining algorithms to one or several streams

Constraints

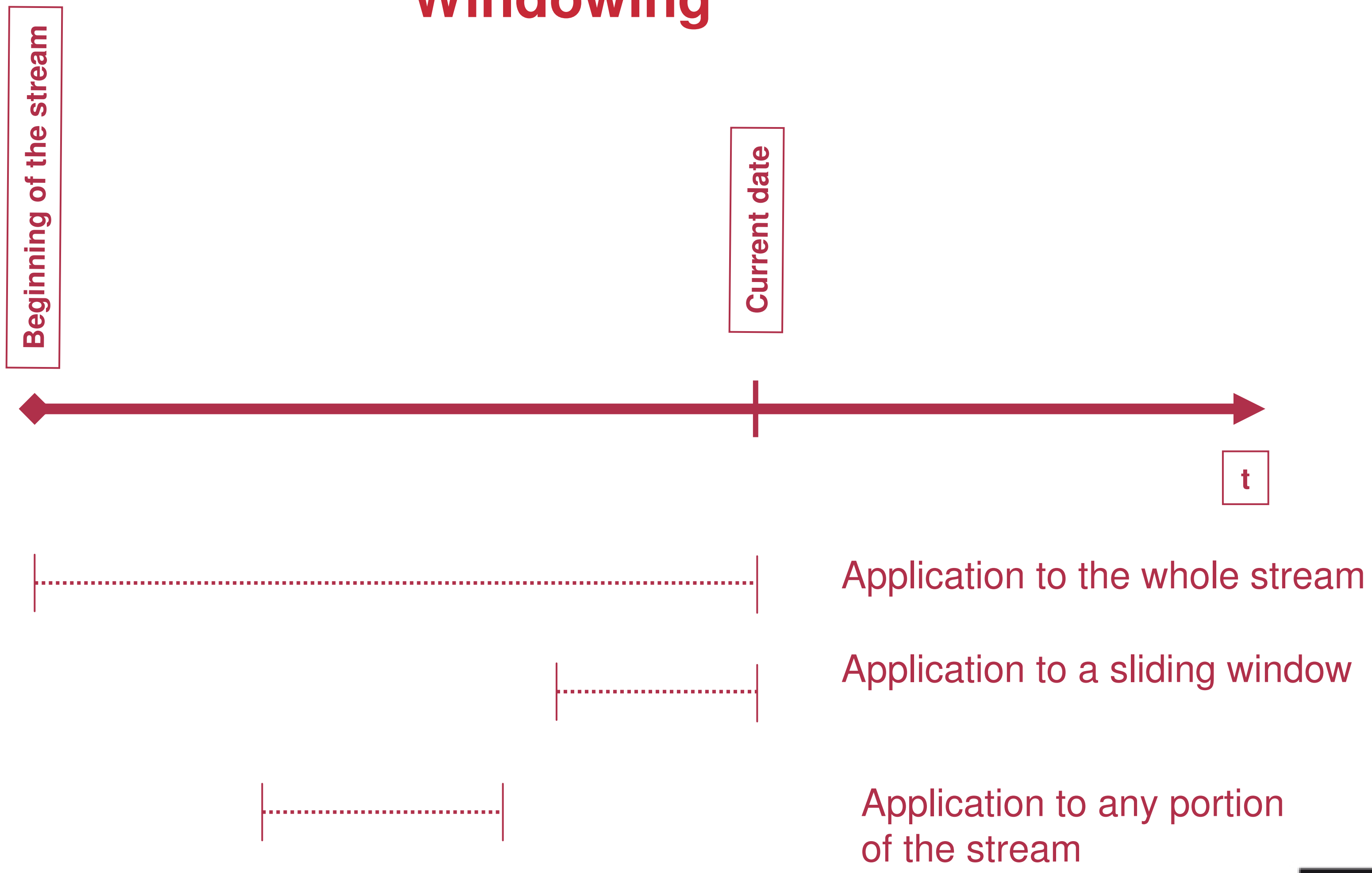
- Limited memory
- Limited CPU
- One-pass

Windowing

TELECOM ParisTech

Data stream mining: definition

Windowing



Data stream mining: definition

Windowing

- Whole stream *(assumes no concept drift)*

  →incremental algorithms

- Sliding window

  →incremental algorithms + ability to forget the past

- Any past portion

  →incremental algorithms + conservation of summaries

TELECOM ParisTech

Data stream mining: definition

Whole stream

- Neural networks
- Adaptation of decision trees

Sliding window

- Additive methods: ex. PCA

Any portion of the stream

- Temporal summaries: CLUSTREAM

Data stream mining outline

- Definition
- Decision tree
- PCA
- Clustream

Data stream mining: decision tree

Adaptation of decision trees to streams

VFDT: **Very Fast Decision Trees (Domingos & Hulten 2000)**

- X_1, X_2, …, X_p: discrete or continuous attributes
- Y: discrete attribute to predict
- Elements of the stream (x_1, x_2, …, x_p, y) are examples
- G(X): measure to maximize to choose splits (ex. Gini, entropy, …)

Data stream mining: decision tree

Hoeffding trees

Idea: *not necessary to wait for all examples to choose a split*

- Minimum number of examples
- Hyp:
  - *$G(X_j)$* can be computed as the mean of values of each example
  - Stable distribution, examples arrive randomly

$$\overline{G(X_j)} \xrightarrow[n \to +\infty]{} G(X_j)$$

$$if \quad \overline{G(X_j)} - \overline{G(X_{j'})} \geq \varepsilon \qquad with \qquad \varepsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}$$

$$then \quad P(G(X_j) > G(X_{j'})) = 1 - \delta$$

Data stream mining: decision tree

Hoeffding trees

Algorithm

- Maintain $G(X_j)$
- Wait for a minimum number of examples
- j, k the 2 variables with highest values of G
- Split on X_j when $G(X_j) - G(X_k) \geq \varepsilon$
- Recursively apply the rule by pushing new examples in the tree leaves

- Sufficient statistics: n_{ijkl} # of items with value i of variable j in class k for leaf l
- VFDT: refinements on this algorithm

Data stream mining outline

- Definition
- Decision tree
- → PCA
- Clustream

Data stream mining: additive methods

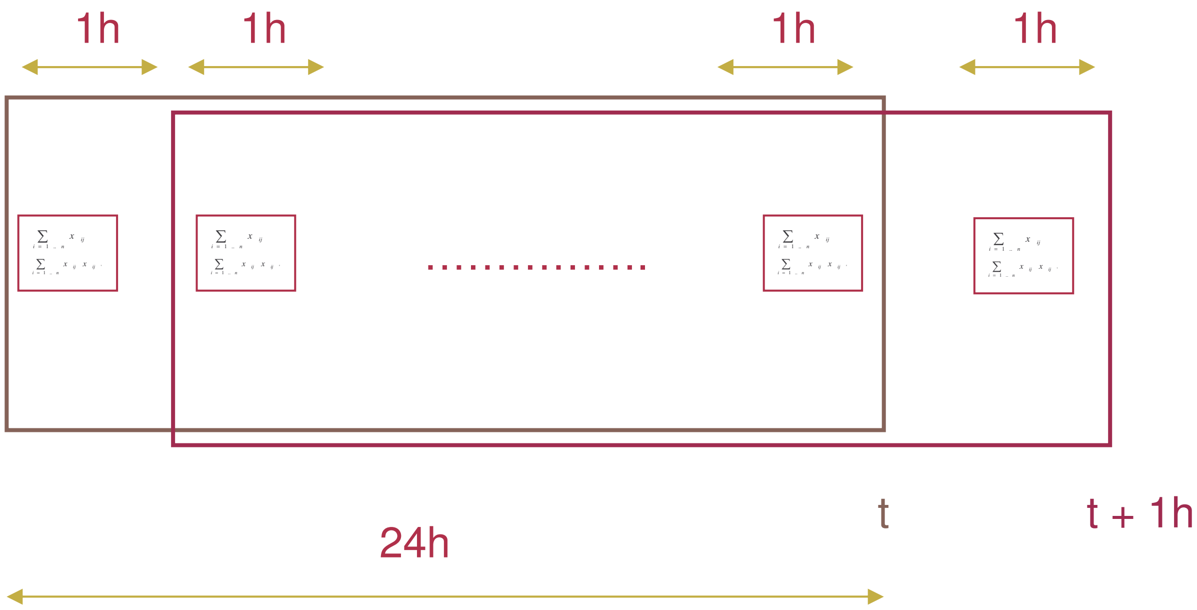
Additive methods: the example of PCA

- Principal Component Analysis
- Items are elements $(x_1, x_2, \ldots, x_n)$ of R^p
- Covariance/correlation matrix $p \times p$
- Incremental maintenance of $p(p+1)$ statistics:

$$\sum_{i=1..n} x_{ij} \qquad \sum_{i=1..n} x_{ij} x_{ij'}$$

- Recomputation of PCA at refreshment rate

Data stream mining: additive methods



Sliding window of 24h

Refreshment every 1h

Data stream mining outline

- Definition
- Decision tree
- PCA
- Clustream

Data stream mining: Clustream

- Summarizing with evolving micro-clusters

- Supports concept drift

- Clustream (Aggarwal et al. 03)
  - Numerical variables
  - Maintenance of a large number of micro-clusters
  - Mecanism to keep track of micro-clusters history

Data stream mining: Clustream

- **Representation of micro-clusters**
  - CVF: Cluster Feature Vector

(n, CF1(T), CF2(T), CF1(X1), CF2(X1), …, CF1(Xp), CF2(Xp))

$$CF1(X_j) = \sum_{i=1..n} x_{ij}$$

$$CF2(X_j) = \sum_{i=1..n} x_{ij}^2$$

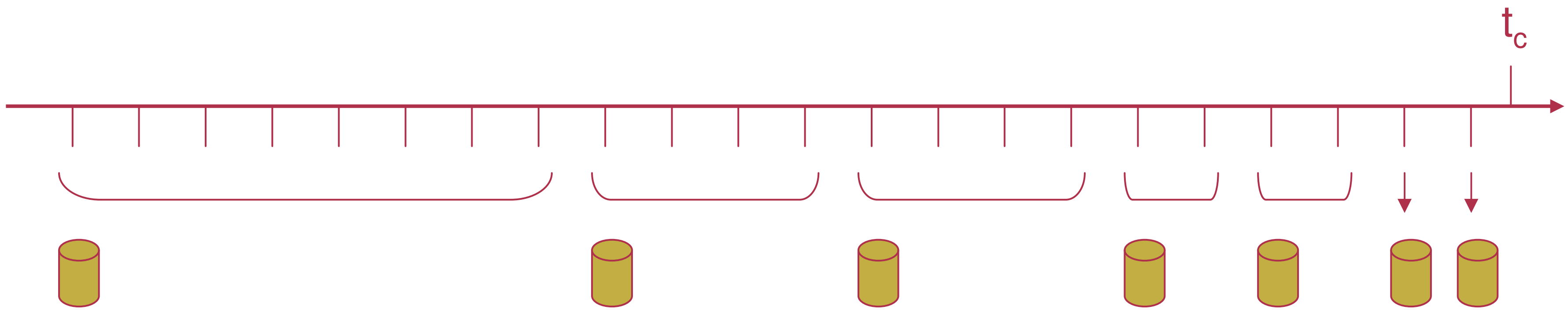
  - Supports union/difference by addition/substraction
  - Incremental computation (elements are disgarded)

Data stream mining: Clustream

- **Maintenance of micro-clusters**
  - Fixed number of micro-clusters
  - Initial micro-clusters (off-line)
  - Each new item:
    - Find closest micro-cluster
    - 'affectation' to a cluster and update of CFV
    - Creation of a new micro-cluster (deletion or merge to make room)
  - List of items of each micro-cluster not maintained
  - History of micro-clusters fusions kept

Data stream mining: Clustream

- **Mecanism to keep track of micro-clusters history**
  - Snapshots at regular time intervals
  - Logarithmic storage structure (bounded)
  - Tilted time windows



Data stream mining: Clustream (Aggarwal et al. 03)

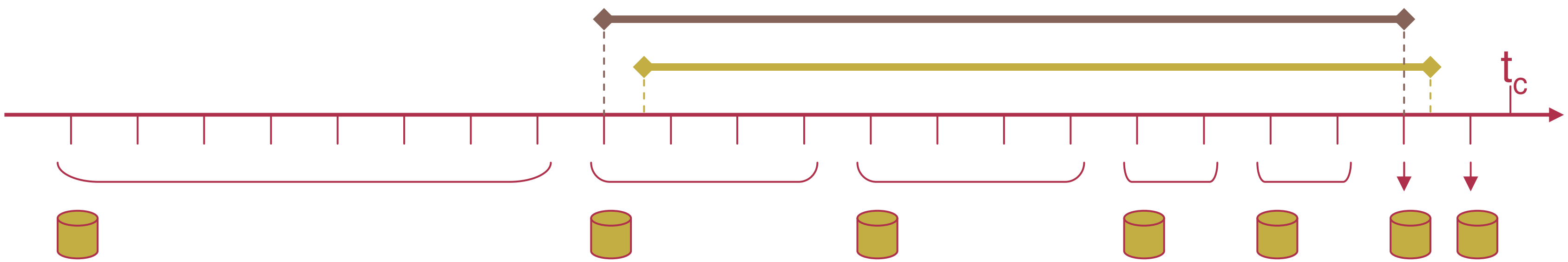
End-user clustering

Selection of relevant data for the period

- Reconstitution of micro-clusters from any past portion
- Use addition/substraction properties of micro-clusters

Hierarchical clustering of micro-clusters

- Standard clustering with weights



Outline

- What is a data stream ?
- Applications of data stream processing
- Models for data streams
- Data stream management systems
- Data stream mining
- Synopses structures
- Conclusion

Synopses structures

Motivation

- Keeping track of a maximum of items in bounded space
- Some operations may still be long even with windowing

→Approximate result based on summarized information

Several approaches

- Random samples
- Histograms
- Sketches

TELECOM ParisTech

Synopses structures: random samples

Problem: **maintain a random sample from a stream**

'Reservoir' sampling **(Vitter 85)**

- Random sample of size *M*
  - Fill the reservoir with the first *M* elements of the stream
  - For element *n* *(n > M)*
    - Select element *n* with probability *M/n*
    - If element *n* is selected pick up randomly an element in the reservoir and replace it by element *n*

Random sampling from a sliding window:
'Chain' sampling (Babcock et al. 2002)

Synopses structures

Motivation

- Keeping track of a maximum of items in bounded space
- Some operations may still be long even with windowing

→Approximate result based on summarized information

Several approaches

- Random samples
- Histograms
- Sketches

Synopses structures: sketches

Sketch

- Synopsis structure taking advantage of high volumes of data
- Provides an approximate result with probabilistic bounds
- Random projections on smaller spaces (hash functions)

Many sketch structures: **usually dedicated to a specialized task**

Examples of sketch structures

- **COUNT** (Flajolet 85)
- **COUNT SKETCH** (Charikar et al. 04)
- → **COUNT MIN SKETCH** (Cormode and Muthukrishnan 03)

Synopses structures: sketches

COUNT MIN SKETCH (Cormode and Muthukrishnan 04)

- *n* observed objects (ex: *n* IP addresses) – *n* very large
- Signal of interest over objects: $a_1(t)$, $a_2(t)$, …., $a_n(t)$
  (ex: # connections)

- Stream contents: (i_t, c_t) with $c_t \geq 0$

$$a_i(t) = a_i(t-1) + c_t \qquad \text{if } i_t = i$$
$$a_i(t) = a_i(t-1) \qquad \text{if } i_t \neq i$$

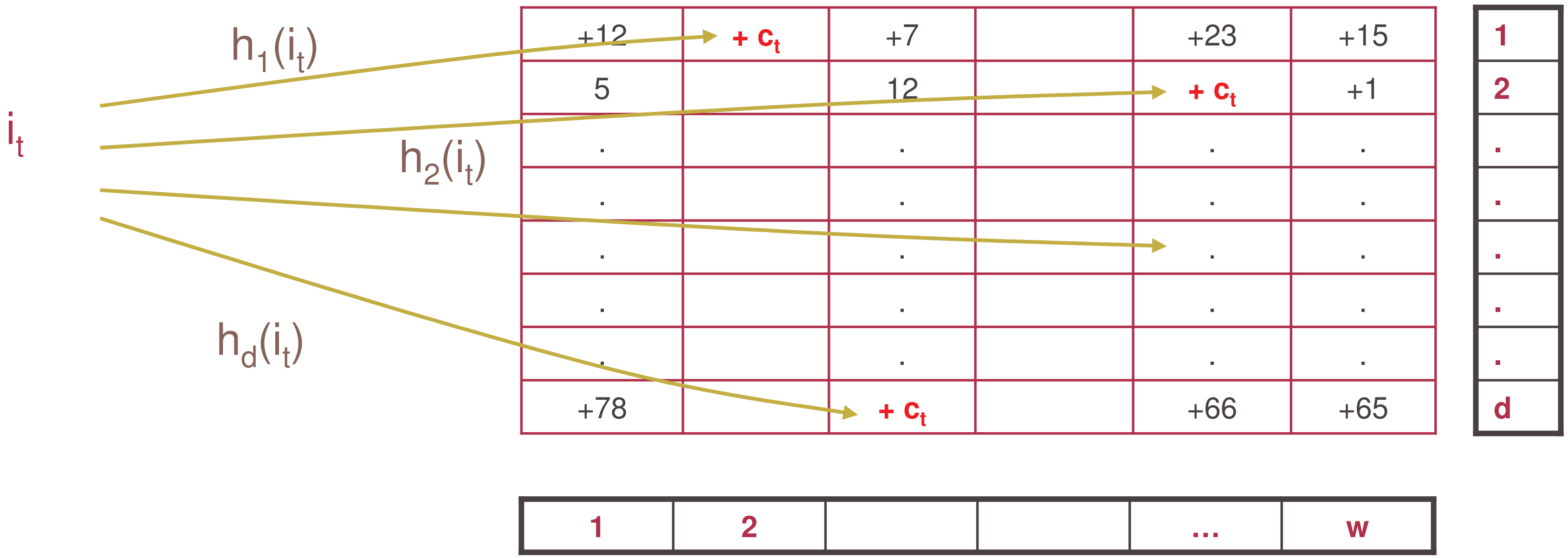
- Queries: $a_i(t)$ for a given *i*
  (ex: # of connections for a given IP address)

TELECOM ParisTech

Synopses structures: sketches

- *d* pair-wise independent hash functions: *{1, …, n}* → *{1, …, w}*
- Array CM of size *d x w*

$$CM[j, h_j(i_t)] \leftarrow CM[j, h_j(i_t)] + c_t$$



- Estimation of $a_i(t) = \min_{j=1..d} (CM[j, h_j(i)])$

Synopses structures: sketches

Bounds on the estimation:

$$0 \leq \hat{a}_i - a_i \leq \varepsilon \|a\|_1 \quad with \quad probability \quad at \quad least \quad 1-\delta$$

$$where \begin{cases} \varepsilon = e / w \\ \delta = e^{-d} \\ \|a\|_1 = \sum_{i=1}^{n} |a_i| \end{cases}$$

Outline

- What is a data stream ?
- Applications of data stream processing
- Models for data streams
- Data stream management systems
- Data stream mining
- Synopses structures
- Conclusion

Conclusion

Very active area of research

Many practical applications in various domains

DSMS are more mature than data stream mining

DSMS

- Commercial efficient systems
- Event processing systems
- Distributed DSMS

Data stream mining

- Already several results
- Still much work to do:
  - Identification and modeling of concept drift
  - Summarizing data stream history (also for DSMS)
  - Distributed data stream mining

Conclusion

French ANR MIDAS project (2008-2010) http://midas.enst.fr

- Generic summaries of data streams
  - Enables queries/mining tasks on any historical part of the stream
  - Several approaches: *sampling, micro-clustering, sequential patterns, automata, OLAP data cubes*

- Applications
  - Utilities: electric power consumption, supervision of power plants
  - Telecommunications: analysis of usage of telecommunication and web services
  - Medical care: monitoring of patients on a hospital
  - Tourism: analysis and recommendation from GPS positions of vehicules

- Partners
  - TELECOM ParisTech, INRIA, LIRMM, CEREGMIA, EDF R&D, Orange Labs

Quiz

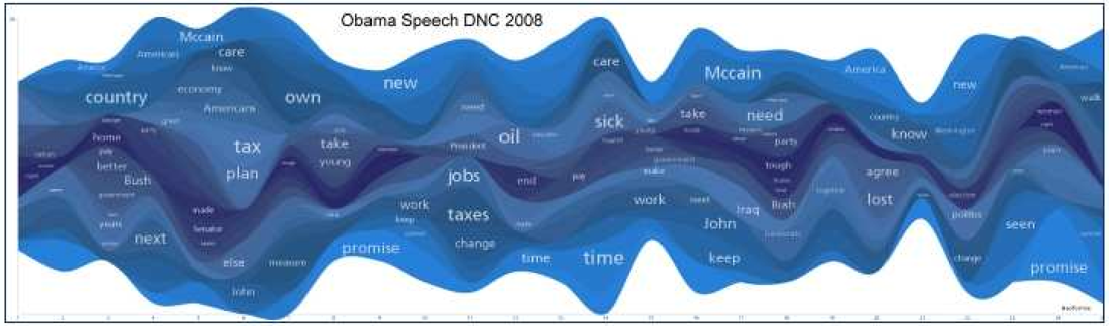
Neoformix

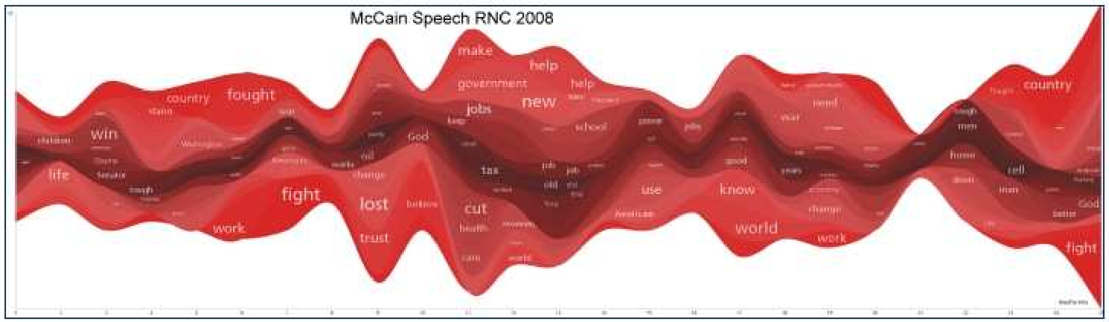
Discovering and Illustrating Patterns in Data

Obama McCain Convention Speech Comparison

By: Jeff Clark Date: Fri, 05 Sep 2008

I have built some graphics comparing the speech delivered by McCain at the RNC last night with the speech from last week by Obama. To start with, here are the StreamGraph diagrams for both speeches. Click on either one to see more detail.





- Home
- About
- Archive
- Portfolio
- Contact
- XML

References: general

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, 2002.

Data streams: algorithms and applications. S.Muthukrishnan, In Foundations and Trends in Theoretical Computer Science, Volume 1, Issue 2, August 2005.

Data streams: models and algorithms. C.C.Aggarwal. Springer, 2007.

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/

TELECOM ParisTech

References: DSMS

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

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. Available at: 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/

Aurora: A New Model and Architecture for Data Stream Management. D. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, S. Zdonik, In VLDB Journal (12)2: 120-139, August 2003.

Load Shedding for Aggregation Queries over Data Streams. B.Babcock, M.Datar, R.Motwani, 2004. Available at:http://www-db.stanford.edu/stream

Aleri software, http://www.aleri.com

Coral8 software, http://www.coral8.com

Streambase software, http://www.streambase.com

References: data stream mining

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.

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.

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.

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 SODA, 2002.

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

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

An improved data stream summary: the count-min sketch and its applications. G.Cormode, S.Muthukrishnan, Journal of Algorithms, Vol.55 , N°1, Academic press, April 2005.

Mining data streams: a review. M.M.Medhat, A.Zaslavsky and S.Krishnaswamy, in SIGMOD Record, Vol.34, N°2, pp.18-26, June 2005.

QUESTIONS ?

Applications of data stream processing

Standard data processing
versus data stream processing

	Standard data processing technology	Data stream processing technology
Monitoring, Business Intelligence applications	**Data warehouses (unscalable)**	**Querying and mining ‘on the fly’ (scalable)**
Applications with basic streaming data	**Specific development without database technology**	**Generic tools for processing data**

Queries in a DSMS

- Main querying approaches for continuous queries
  - Graphical combination of operators on streams

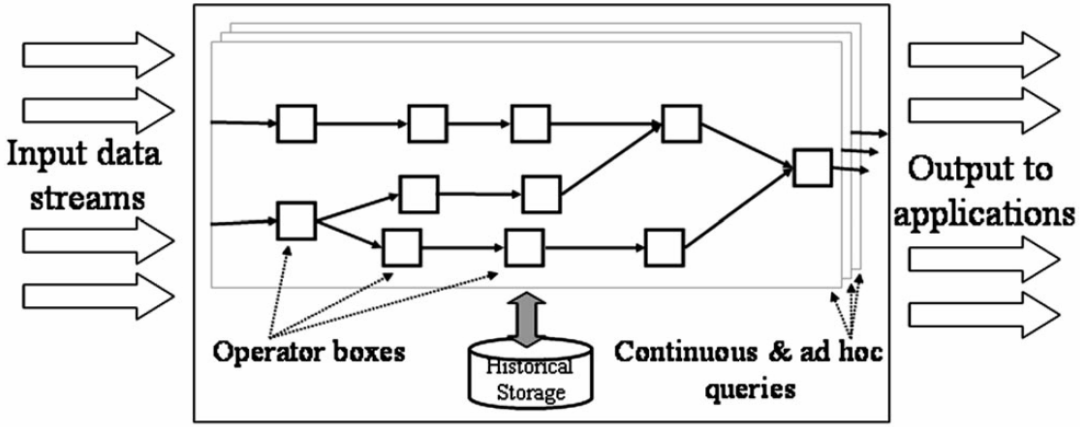


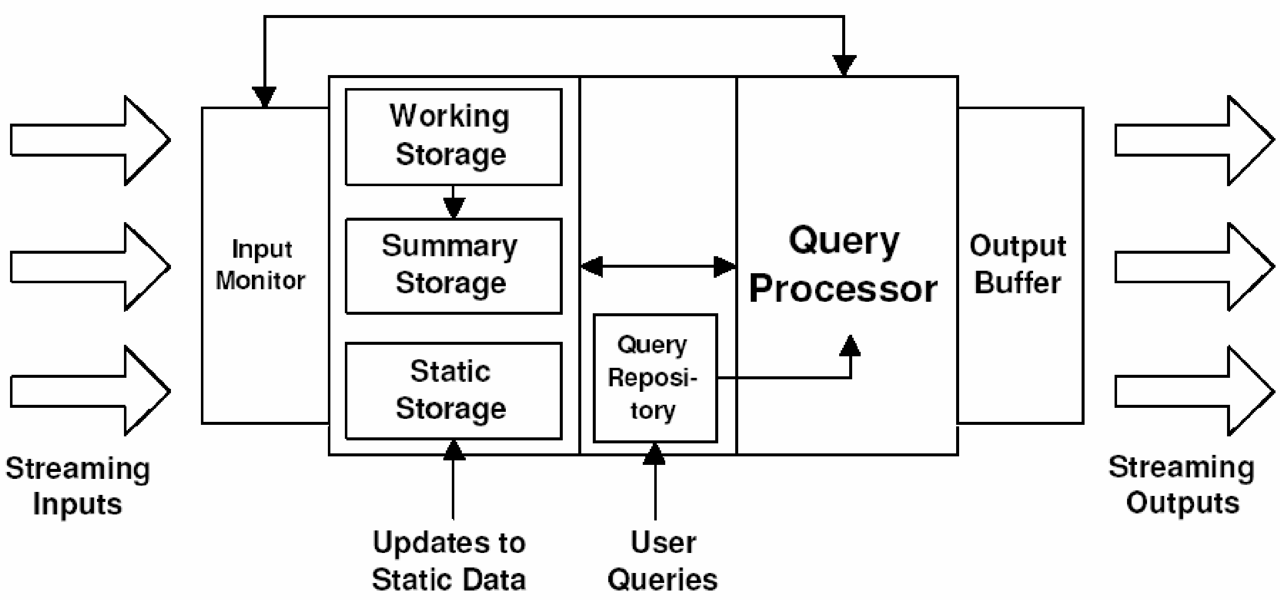
Fig. 1. Aurora system model

Source: Aurora: a new model and architecture for data stream management, VLDB Journal 2003

  - Extensions of SQL to continuous queries: the STREAM project

Approximate answers to queries

- One generic architecture proposed by Golab et Ozsu (2003):



Source: Golab & Özsu 2003

Approximate answers to queries

Load shedding

- Goal
  - Face (dynamically) high arrival rates in streams by sampling tuples
  - Control the error using a quality of service function
- Principle
  - Set sampling operators in the data flow diagram
  - Optimize dynamically the location/rate of sampling operators

Approximate answers to queries

Example of load shedding approach:

Babcock, Datar and Motwani (STREAM Project)

- Aggregate queries:
  - SUM, COUNT
  - Intermediate selections
  - External joins with fixed relations by foreign keys

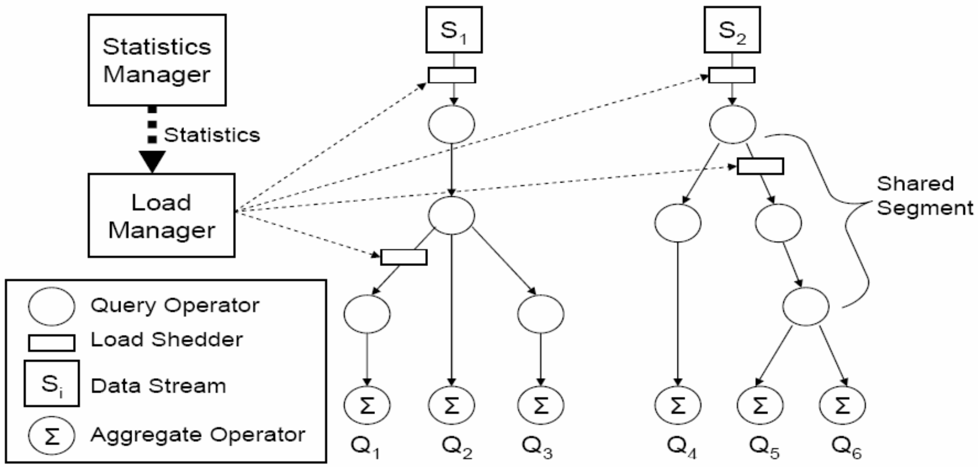
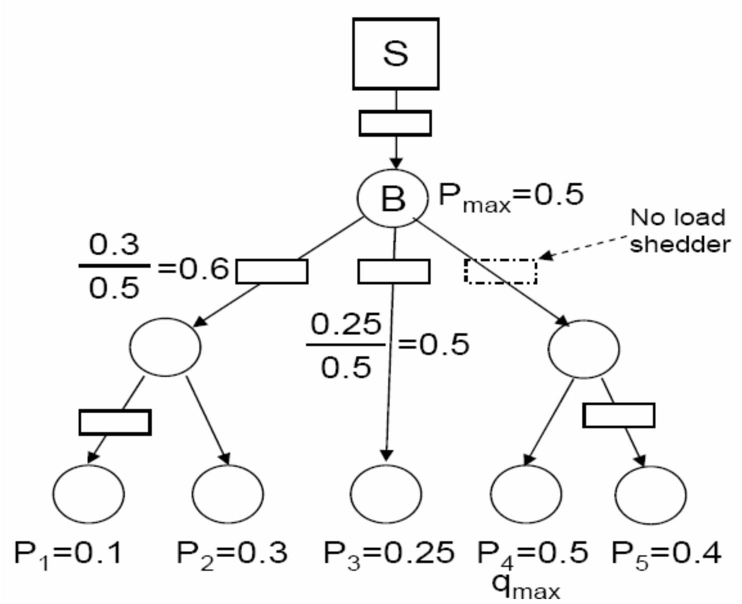


Figure 1. Data Flow Diagram

Approximate answers to queries

Parameters of the problem

- For each operator O_i : selectivity s_i, processing time of a tuple t_i
- For each terminal operator (SUM) : result average μ_i and standard-deviation σ_i
- For each stream: r_i arrival rate of tuples
- For each operator O_i : p_i is the number of tuples to send to it by unit of time



Problem definition

- Determine p_i's by minimizing the maximum error on terminal operators under the constraint of system max load

Synopses structures: sketches

COUNT **(Flajolet 85)**

Goal

- Number N of distinct values in a stream (for large N)
- Ex. number of distinct IP addresses going through a router

Sketch structure

- SK: L bits initialized to 0

0	0	0	0	0	0	0	0

- H: hashing function transforming an element of the stream into L bits

18.6.7.1 $\longrightarrow$

0	0	1	0	1	0	1	0

- H distributes uniformly elements of the stream on the 2^L possibilities

TELECOM ParisTech

Synopses structures: sketches

Method

- Maintenance and update of SK
  - For each new element e
  - Compute H(e)
  - Select the position of the leftmost 1 in H(e)
  - Force to 1 this position in SK

SK	0	1	0	0	1	0	0	1
H(18.6.7.1)	0	0	1	0	1	0	1	0
New SK	0	1	1	0	1	0	0	1

Synopses structures: sketches

Result

- Select the position *R (0…L-1)* of the leftmost 0 in SK
- $E(R) = \log_2 (\varphi^* N)$ with $\varphi = 0.77351\ldots$
- $\sigma(R) = 1.12$

SK	1	1	1	0	1	0	0	0
				↑ R				

For *n* elements already seen, we expect:

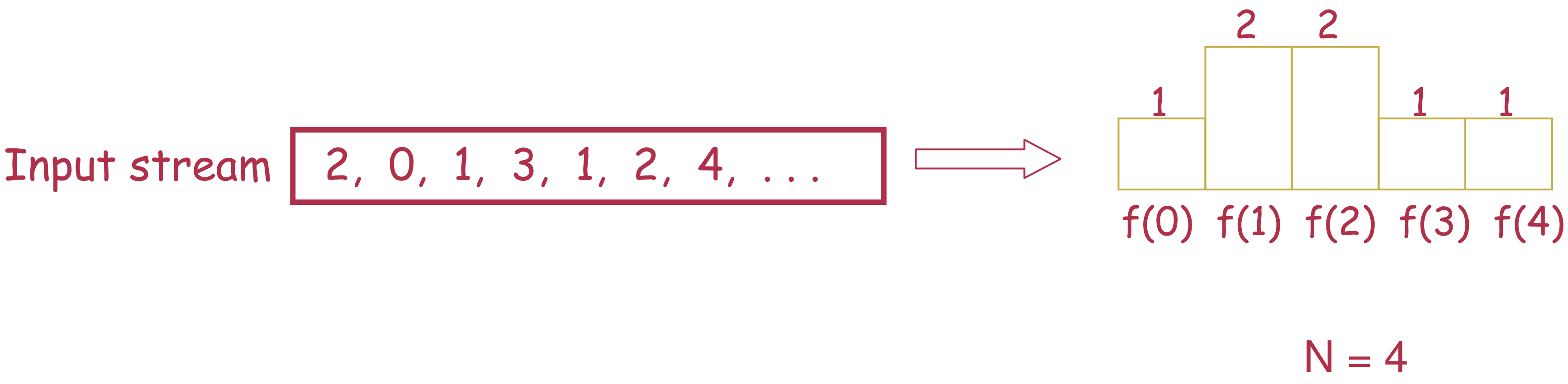
- SK[0] is forced to 1 N/2 times
- SK[1] is forced to 1 N/4 times
- SK[k] is forced to 1 $N/2^{k+1}$ times

Synopses structures: sketches

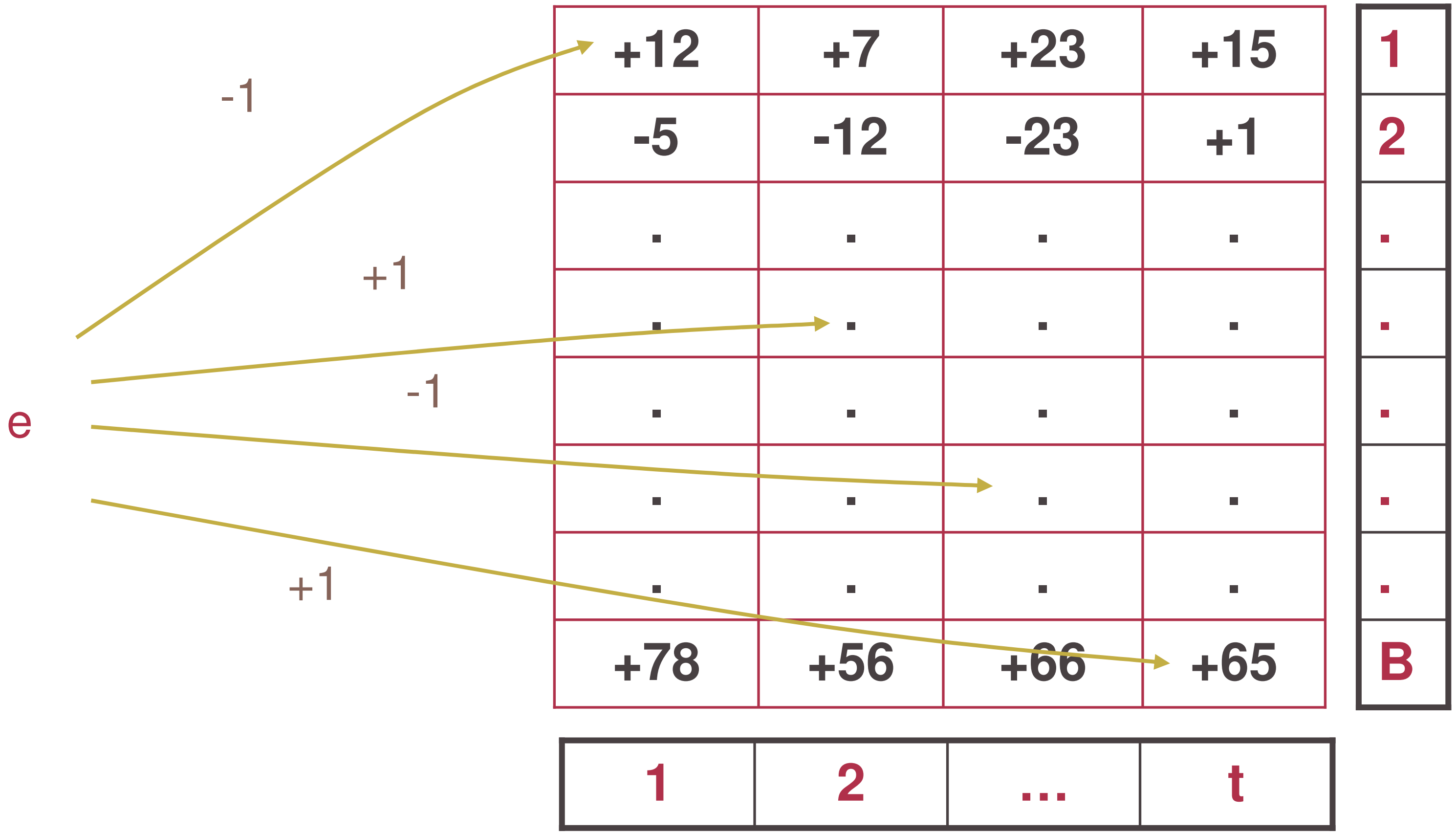
COUNT SKETCH ALGORITHM **(Charikar et al. 2004)**

Goal

- *k* most frequent elements in a stream (for large number *N* of distinct values)
- Ex. 100 most frequent IP addresses going through a router



Synopses structures: sketches



Synopses structures: sketches

Sketch structure

- h : hash function from [0, … , N-1] to [0, 1, … , B]
- s : hash function from [0, … , N-1] to {+1, -1}
- Array of B counters: $C_1, \ldots, C_B$ (with $B << N$)

Sketch maintenance

- when e arrives: $C_{h(e)} \mathrel{+}= s(e)$

Use of sketch

- Estimation of frequency of object e: $n_e \approx C_{h(e)} \,.\, s(e)$
- Actually t hash function h and t hash function s:

$$n_e \approx \text{median}_{\, j \in [1 \ldots t]} \left(C_{hj(e)} \,.\, s_j(e) \right)$$

- Theoretical results on error depending on N, t and B.

Synopses structures: sketches

Algorithm

Maintenance of a list $(e_1, e_2, \ldots, e_k)$ of the current k most frequent elements

For a new arriving element e

- Add e to the sketch structure
- Estimate frequency of e from the sketch structure
- If $f(e) > f(e_k)$, remove e_k and insert e into the list