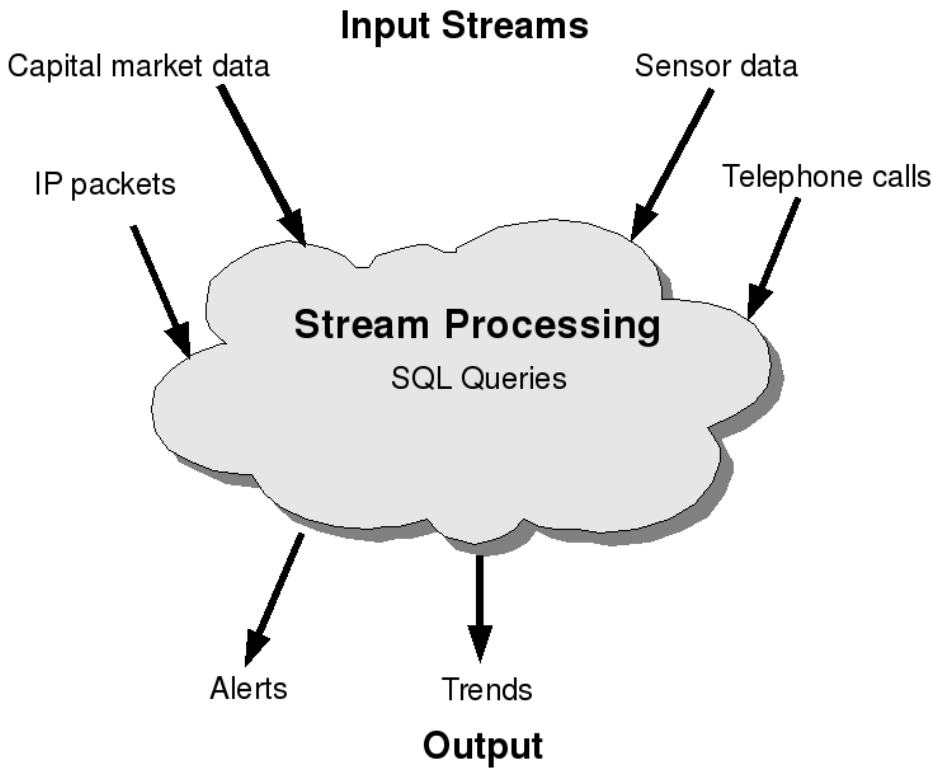
Data Stream Management Systems

Dr. Wenceslao PALMA
wenceslao.palma@ucv.cl

March 2011

Introduction

- A data stream is an unbounded sequence of data that arrives at high speed.
- Stream processing applications require continuous and low-latency processing of data streams.
- In differents domains, such as computer networks, web logs, financial services, applications require traditionally the processing of large data streams.
- Real data traces of IP packets from an AT&T data source show an average data rate of approximately 400 Mbits/sec.



Processing a query over a data streams involves:

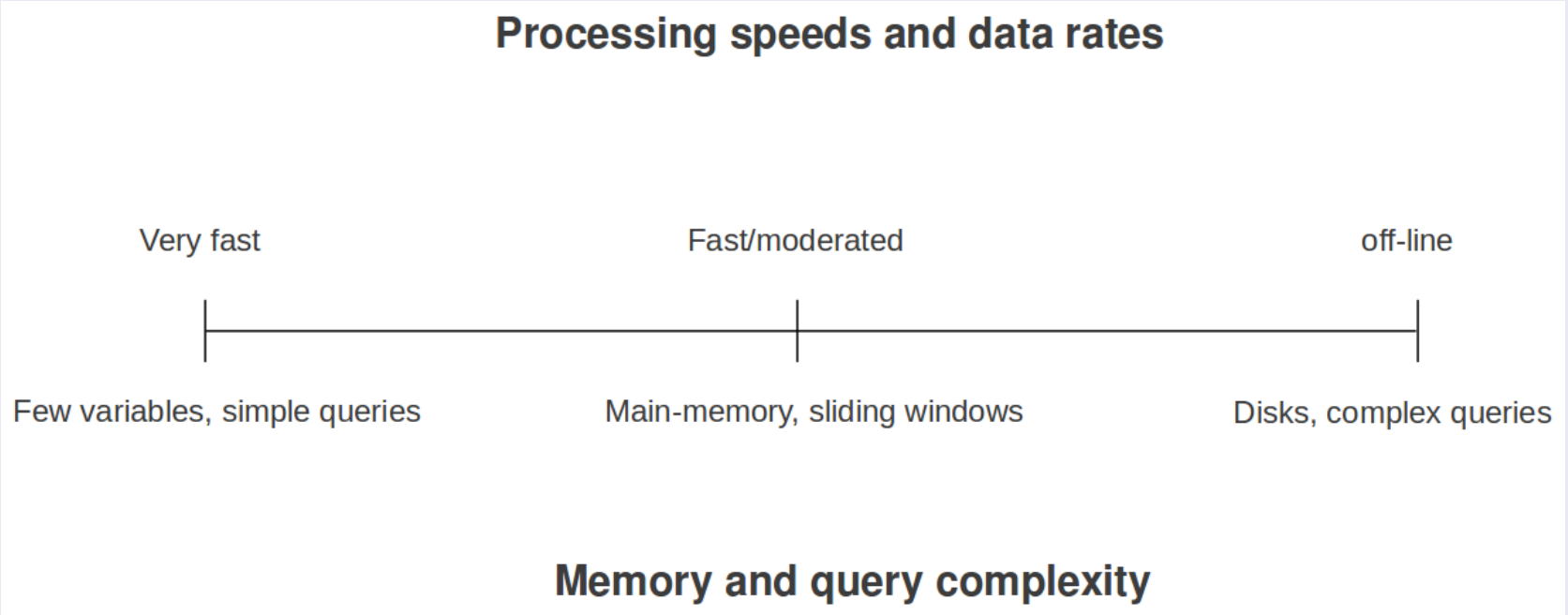
- running the query continuously over the data stream.
- generating a new answer each time a new data item arrives.

Processing a query over a data streams involves:

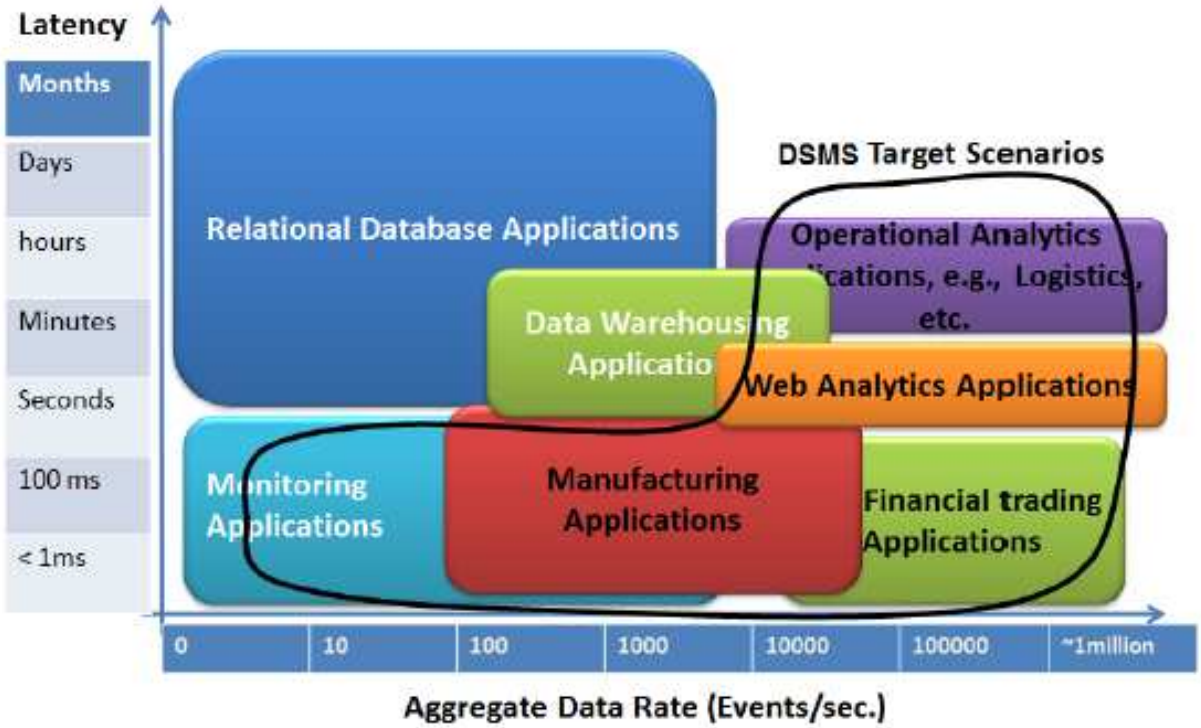
- running the query continuously over the data stream.
- generating a new answer each time a new data item arrives.

The requirements of data stream applications do no fit the DBMS data model and querying paradigm

Application requirements



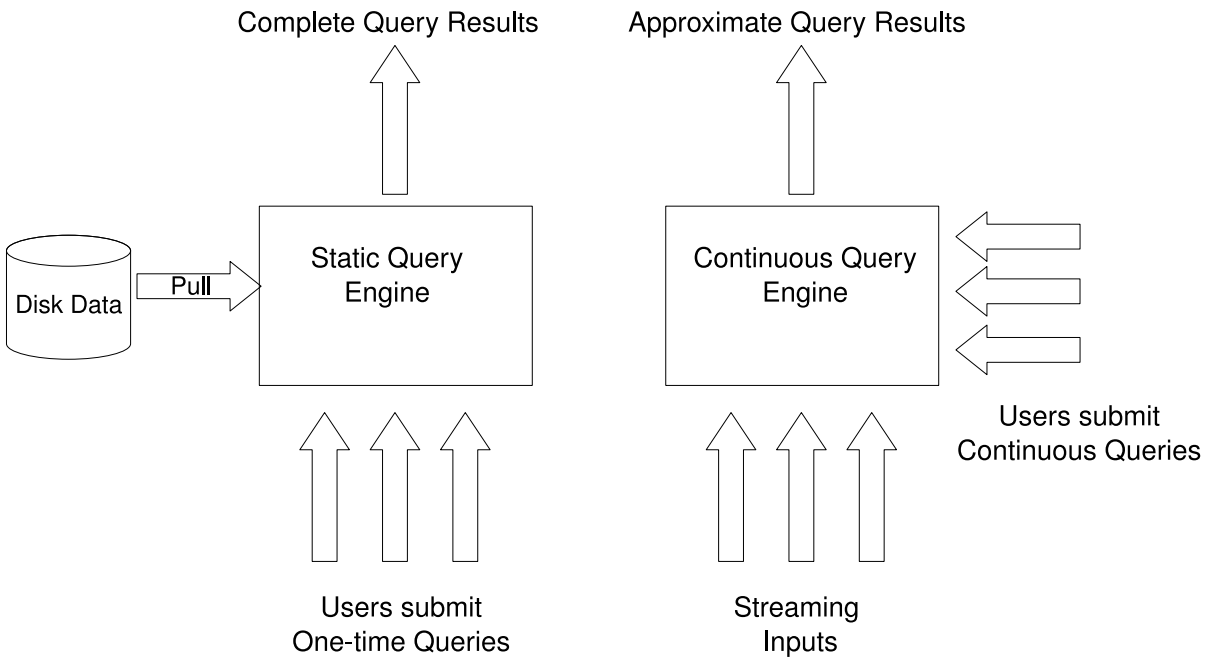
DSMSs scenarios



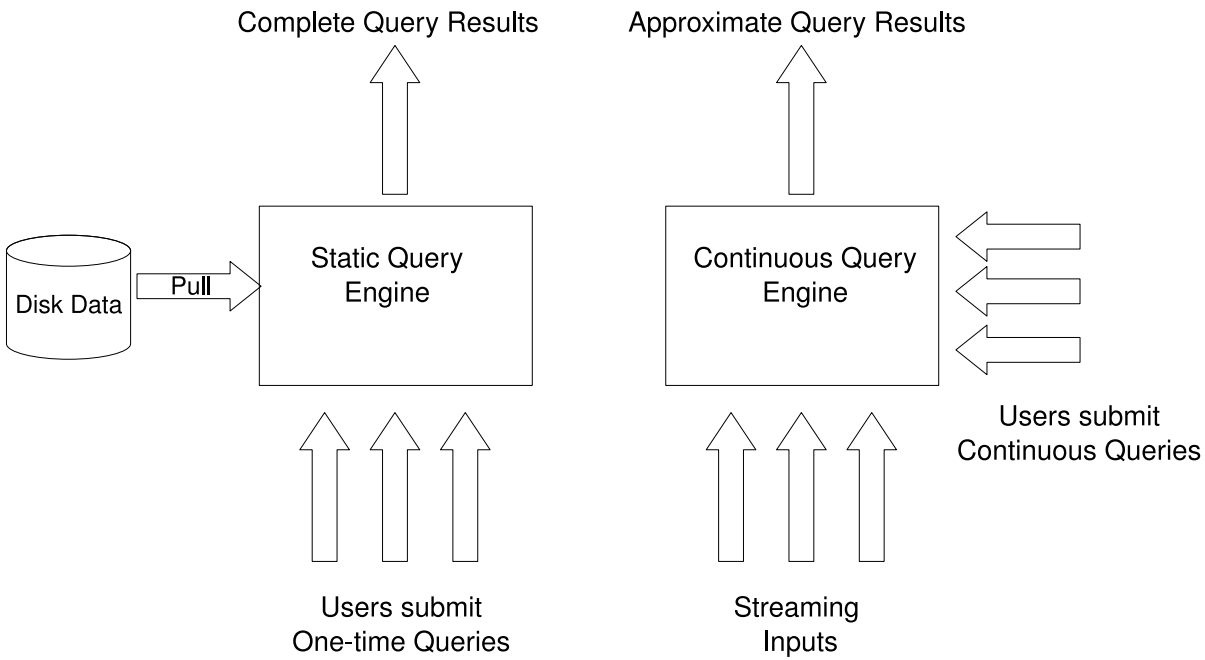
1

[1] taken from paper *Data Stream Management Systems for Computational Finance*

DBMS vs DSMS


Complete Query Results
Approximate Query Results
Disk Data
Pull
Static Query Engine
Continuous Query Engine
Users submit Continuous Queries
Users submit One-time Queries
Streaming Inputs

DBMS vs DSMS

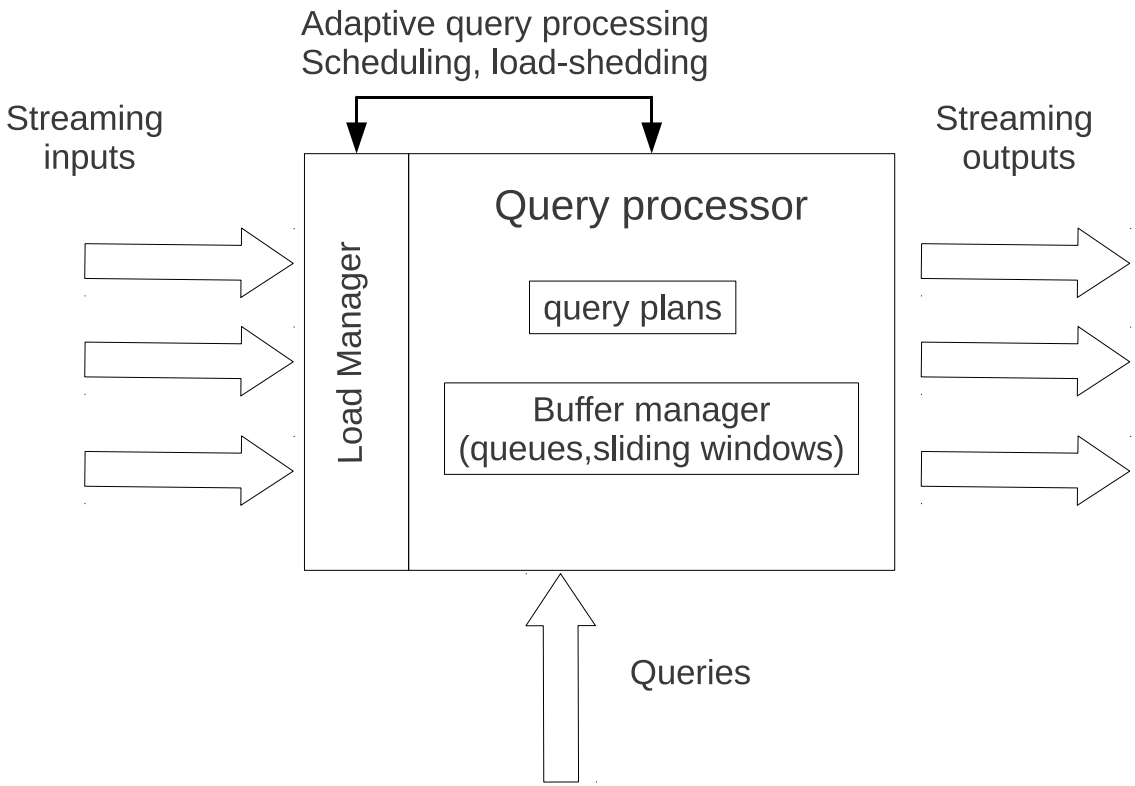


- Persistent queries
- Push-based processing
- Approximate answers

DBMS vs DSMS

	DBMS	**DSMS**
Data	persistent	streams,sliding windows
Data access	random	sequential, one-pass
Updates	arbitrary	append-only
Update rates	slow	high and bursty
Processing model	query-driven	data-driven
Queries	one-time	continuous
Query plans	fixed	adaptive
Query optimization	one-query	multi-query
Query answers	exact	approximate
Latency	relatively high	slow

DSMS architecture



Queries

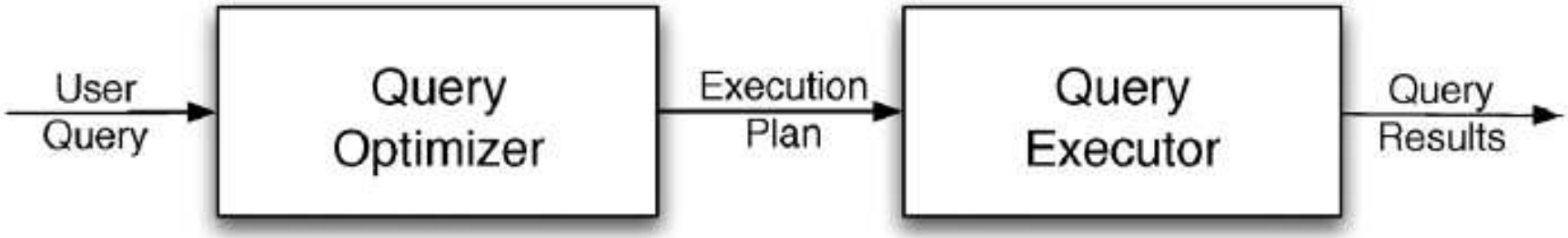
Traffic that passes through three routers R_1, R_2 y R_3 and has the same destination host within the last 10 minutes.

Select sum($S_1.size$)
From S_1[range 10 min], S_2[range 10 min], S_3 [range 10 min]
Where S_1.dest=S_2.dest and S_2.dest=S_3.dest

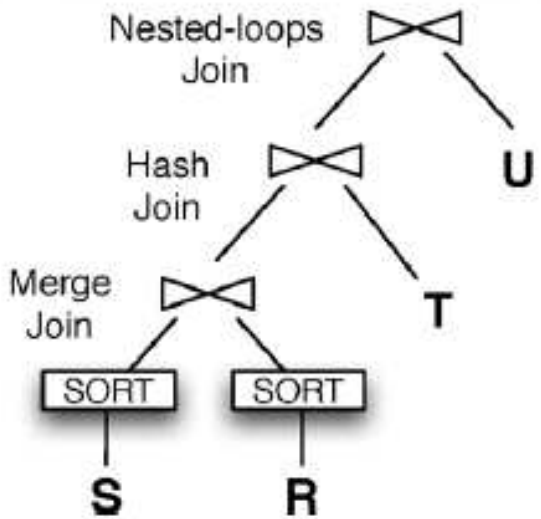
- Data are stored in sliding windows of size $W = 10$.
- Each tuple has a timestamp ts. Thus, a tuple is contained in the window iff $T - s.ts \leq W$.
- Update of tuples is performed by sliding the window $\rightarrow$ the removal of some tuples from the window and the addition of some new tuples arriving in the data streams.

The traditional join query operator has a blocking behaviour because to produce the first result it must see its entire input. Since data streams may be infinite, a blocking operator will never see its entire input not being able to produce any result.

Traditional join operator




```
SELECT *
FROM R, S, T, U
WHERE R.a = S. a
   AND S.b = T.b
   AND T.c = U. c
```

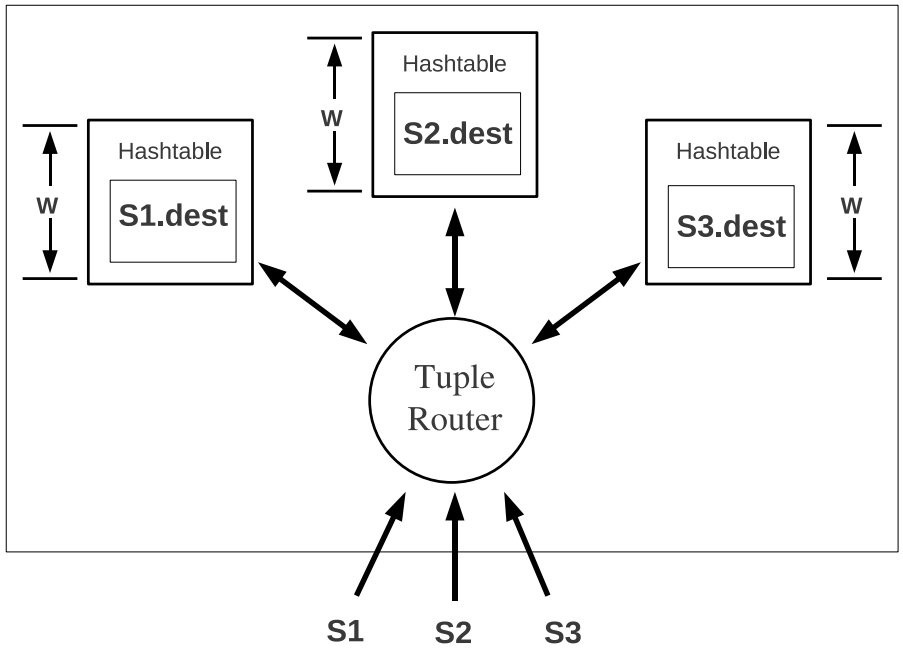


MJoin operator

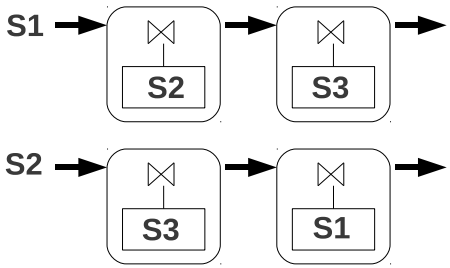
Example 3-way join query

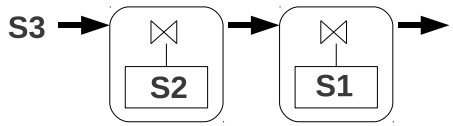
Select *
From S1[range 10 min],S2[range 10 min],S3[range 10 min]
Where S1.dest=S2.dest=S3.dest

MJoin Operator



Example of Probing sequences





Summary of DSMSs and their primary contributions

DSMS	Primary contribution
TelegraphCQ	Operators for adaptive query processing.
STREAM	Adaptive caching for continuous queries and query language.
Borealis	Techniques for fault-tolerance and load management.
DCAPE	Integrates local query optimization and distributed load balancing