Data Stream Management Systems

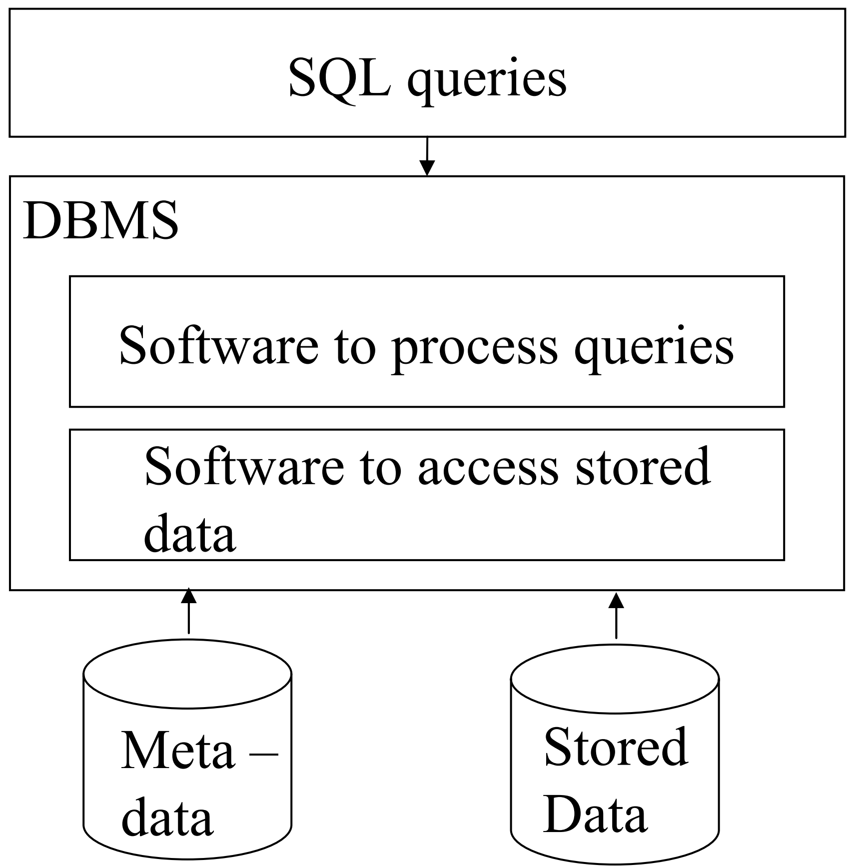
Principles of Modern Database Systems 2007

Tore Risch
Dept. of information technology
Uppsala University
Sweden

Tore Risch
Uppsala University, Sweden

What is a Data Base Management System?

Users and programmers



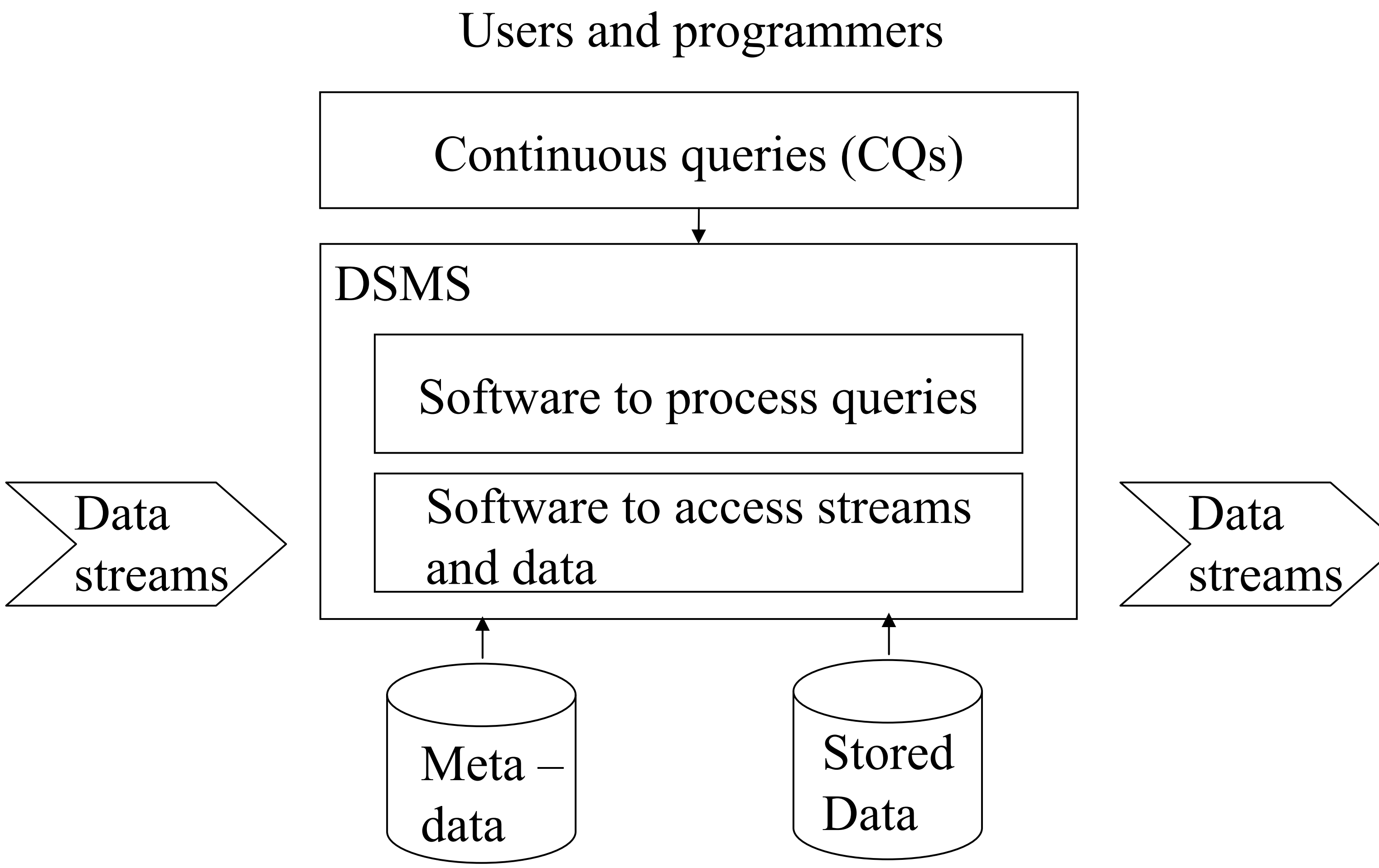
New applications

- Data comes as large data streams, e.g.
- Satellite data
- Scientific instruments
- Colliders
- Patient monitoring
- Stock data
- Process industry
- Traffic control

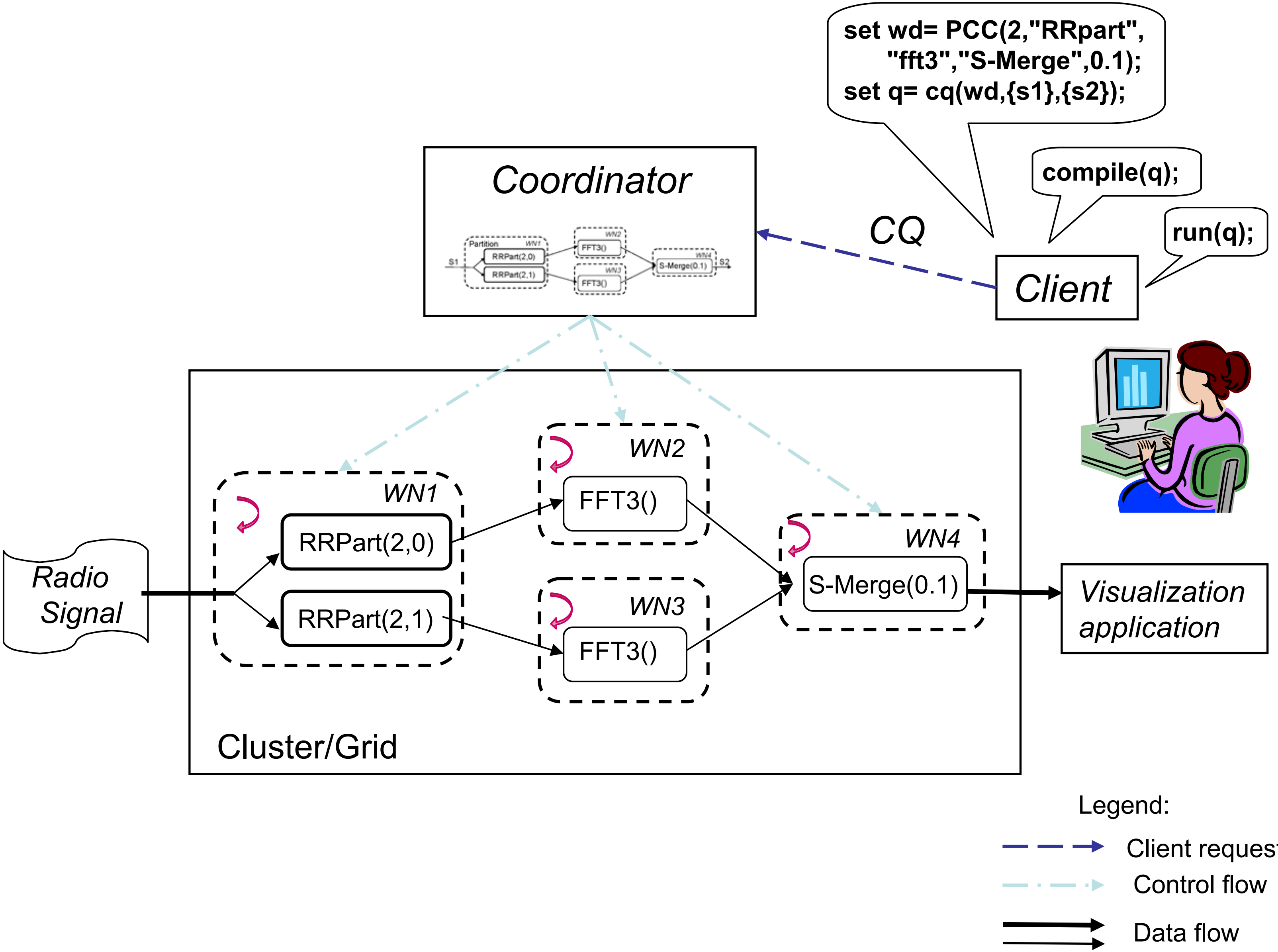
$\Rightarrow$Would like to query data in *streams*

Tore Risch
Uppsala University, Sweden

What is a Data Stream Management System?



DSMS Scenario



Overview paper

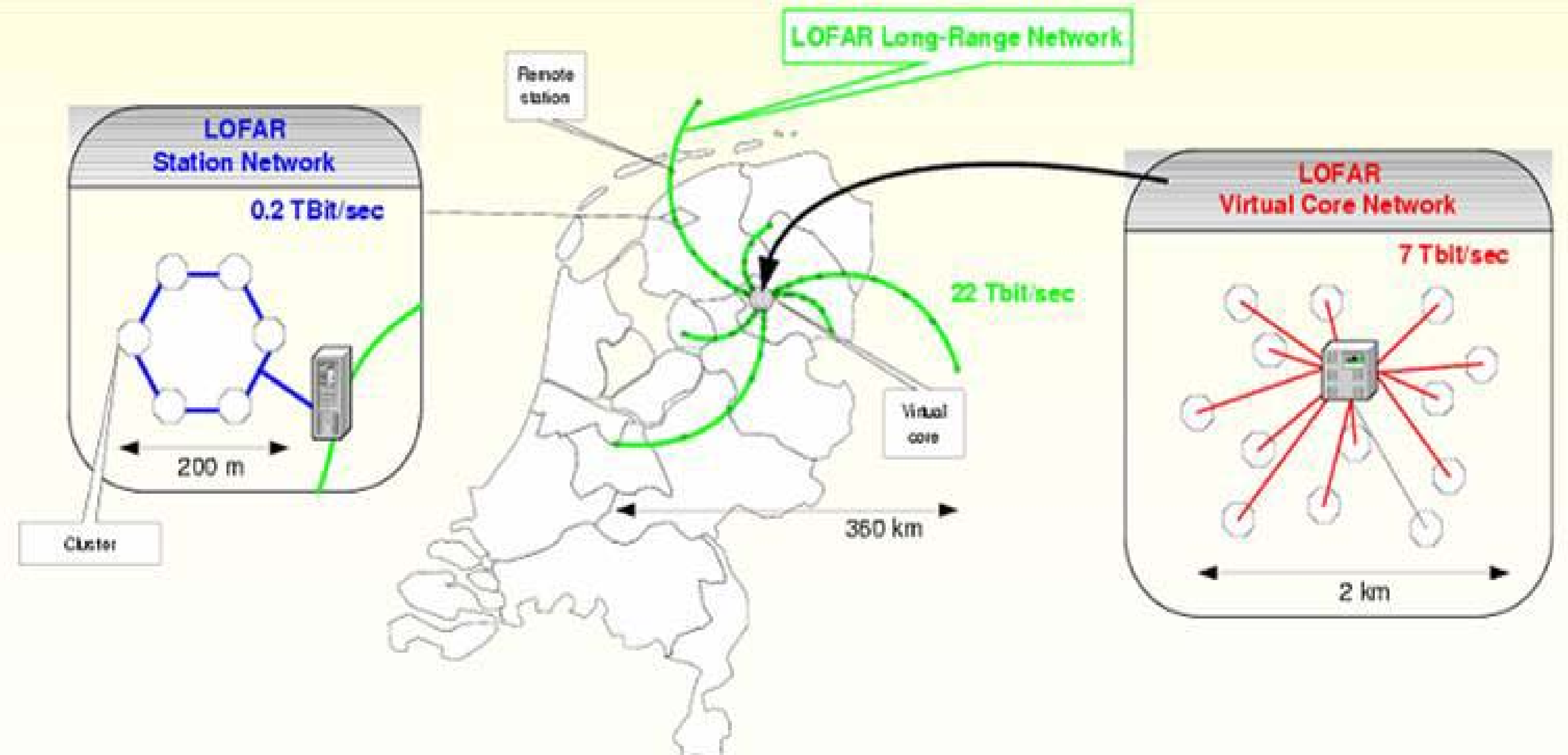
⇒L. Golab and T. Özsu: Issues in Stream Data Management, *SIGMOD Records, 32(2), June 2003, http://www.acm.org/sigmod/record/issues/0306/1.golab-ozsu1.p*

The LOFAR Instrument

-13000 antennas

-Distributed over 100 stations

-Producing ~20Tbps raw data



JU: Developing a scalable DSMS to process LOFAR stream queries

Streams vs tables

- Streams potentially *infinite* in size
  - Regular DBs based on queries to finite tables
- Streams ordered, i.e. *sequence* data
  - Regular DBs are based on sets and bags
- *Stop condition* indicates when/if streams end
- Often very high stream data volume and rate
  - Regular DBs usually less demanding
- Real-time delivery, Quality of Service
  - Regular DBs weak here
- Active query model, *continuous* queries
  - Regular DB queries *passive*

Continuous queries

- CQs are turned on and run until stop condition true
  - Regular queries executed until finished by demand
- CQs return *unbounded* data (streams) as result
  - Regular queries *bounded* by size of tables
- CQs operators usually *montone*, i.e. cannot re-read stream
  - Reqular queries can access same table many times
- CQs specified over *stream windows* (i.e. bounded stream segments)
  - Regular queries specified over entire tables
- CQs often based on time stamps (logs) of stream elements (*temporal*)
  - Regular queries not temporal
- CQ join operators *approximate*
  - Regular join operators usually exactly match data

Stream windows

- Need monotone *window operator* to chop stream into segments
- Window *size* (*sz*) based on:
  - Number of elements
    E.g. last 10 elements
  - Time
    E.g. elements last second
- *Landmark* window:
  - Window from start of stream
  - Continously growing
  - Not bounded
  - Materialization
- Windows also have *stride (str)*
  - Rule for how they move forward

Window stride

- How fast the window moves forward
- *Jumping* window
  $sz = str$
  => Output data rate o = input data rate i
  => No overlap between windows
  => All data processed once
  => C.f. "window rate" $wr=i/sz$
- *Sliding* windows
  $str < sz$
  => $o > i$ $(o = i*sz/str)$
  => Overlaps between windows
  => Data processed more than once
- *Sampling* window
  $str > sz$
  => o < i
  => No overlaps
  => Some data not processed
  => a form of *schredding*

Joining streams

- Streams infinite
  => Monotone join operators needed
  => regular join impossible (not monotone)
- Instead streams are *merged:*
  1. Split stream into segments by *window operator*
  2. Join windows from each stream
  3. Merge the result
- Stream merge is *approximate* join method
  - Window size determines quality of result
- Stream joins need to deal with rate differences, blocking
  => *Time-out* when data blocks
  => *Load shredding* skips stream elements
  => Can also do *approximations* (e.g. aggregation)
  => Need to deal with nulls (c.f. outer joins)

Stream joining methods

- Special join methods different from table joins
- Xjoin:
  T. Urhan and M. Franklin. Dynamic pipeline scheduling for improving interactive performance of online queries.*Proceedings of the VLDB Conference,* 2001.
- Mjoin:
  S. Viglas, J. Naughton, and J. Burger. Maximizing the output rate of multi-join queries over streaming information sources. In *Proc. of the VLDB Conference 2003*
- Hybride:
  Babu, Munagala, Widom, Motwani:Adaptive Caching for Continuous Queries, *Proc. 21st International Conference on Data Engineering (ICDE 2005)*

Punctuations

- Can be seen as corresponding to transactions
- Condition for a unit of work
  E.g. deal is done => new data about it ignored
- Add *punctuation* token in stream
- May improve performance
- Syncronization
- Punctuated joins:

Ding, Mehta, Rundensteiner, Heineman: Joining Punctuated Streams, *EDBT 2004*

DSMS Systems

Aurora (Brown,MIT,Brandeis): Carney et al: Monitoring Streams – A New Class of Data Management Applications, *VLDB 2003*

TelegraphCQ (Berkeley): Chandrasekaran et al: TelegraphCQ: Continuous Dataflow Processing for an Uncertain World, CIDR 2003

Gigascope (AT & T): Cranor et al: Gigascope: High Performance Network Monitoring with an SQL Interface, SIGMOD 2002

STREAM (Stanford):StreaMon: Baby & Widom: An Adaptive Engine for Stream Query Processing, SIGMOD 2004

Borealis (Brown & Brandeis): Ahmad et al: StreaMon: An Adaptive Engine for Stream Query Processing, SIGMOD 2005 (distributed streams)

Wavescope (MIT): Girod et al: The Case for a Signal-Oriented Data Stream Management System, CIDR 2007

Own related efforts

SCSQ (Zeitler & Risch): Processing high-volume stream queries on a supercomputer, ICDE Ph.D. Workshop 2006 (distributed, numerical)

GSDM (Ivanova & Risch): Customizable Parallel Execution of Scientific Stream Queries, VLDB 2005 (distributed, numerical)

L.Lin, T. Risch: Querying Continuous Time Sequences , VLDB 1998 (numerical time series)

Aggregation over stream windows

E.g. SCSQ:

```
select avg(winagg(s,100,30))
 from Stream s
 where id(source(s))=2;
```

- Lots of work on similarity search over time sequences
- Indexing time series

Bulut and Singh: A Unified Framework for Monitoring Data Streams in Real Time, ICDE 2005

Zhu and Shasha: Warping Indexes with Envelope Transforms for Query by Humming, SIGMOD 2003

Scientific Databases

- Optimization of queries with numerical functions

  Wolniewicz and Graefe: Algebraic Optimization of Computations overScientific Databases, VLDB 1999

- Function approximation and caching

  Panda, Riedewald, Pope, Gehrke, Chew: Indexing for Function Approximation, VLDB 2006

  Denny & Franklin: Adaptive Execution of Variable-Accuracy Functions, VLDB 2006

Scientific Databases

- Scientific workflows

  Berkley et al: Incorporating Semantics in Scientific Workflow Authoring, SSDBM 2005

- Tracking changes and sources

  Buneman et al: Provenance Management in Curated Databases, SIGMOD 2006

- Spatial indexing (c.f. multimedia databases)

  Csabail et al: Spatial Indexing of Large Multidimensional Databases, CIDR 2007