



UPPSALA
UNIVERSITET

Customizable Parallel Scientific Stream Processing

Tore Risch, Milena Ivanova,
Erik Zeitler

Uppsala Database Laboratory

<http://user.it.uu.se/~udbl/>



Milena Ivanova and Tore Risch: Customizable Parallel Execution of Scientific Stream Queries, *Proc. 31st International Conference on Very Large Databases, VLDB2005*, Trondheim, Norway, 2005, pp 157-168.

Milena Ivanova: *Scalable Scientific Stream Query Processing*, PhD Thesis, 2005,
<http://user.it.uu.se/~udbl/Theses/MilenaIvanovaPhD.pdf>.



Stream Database Manager GSDM

Milena Ivanova

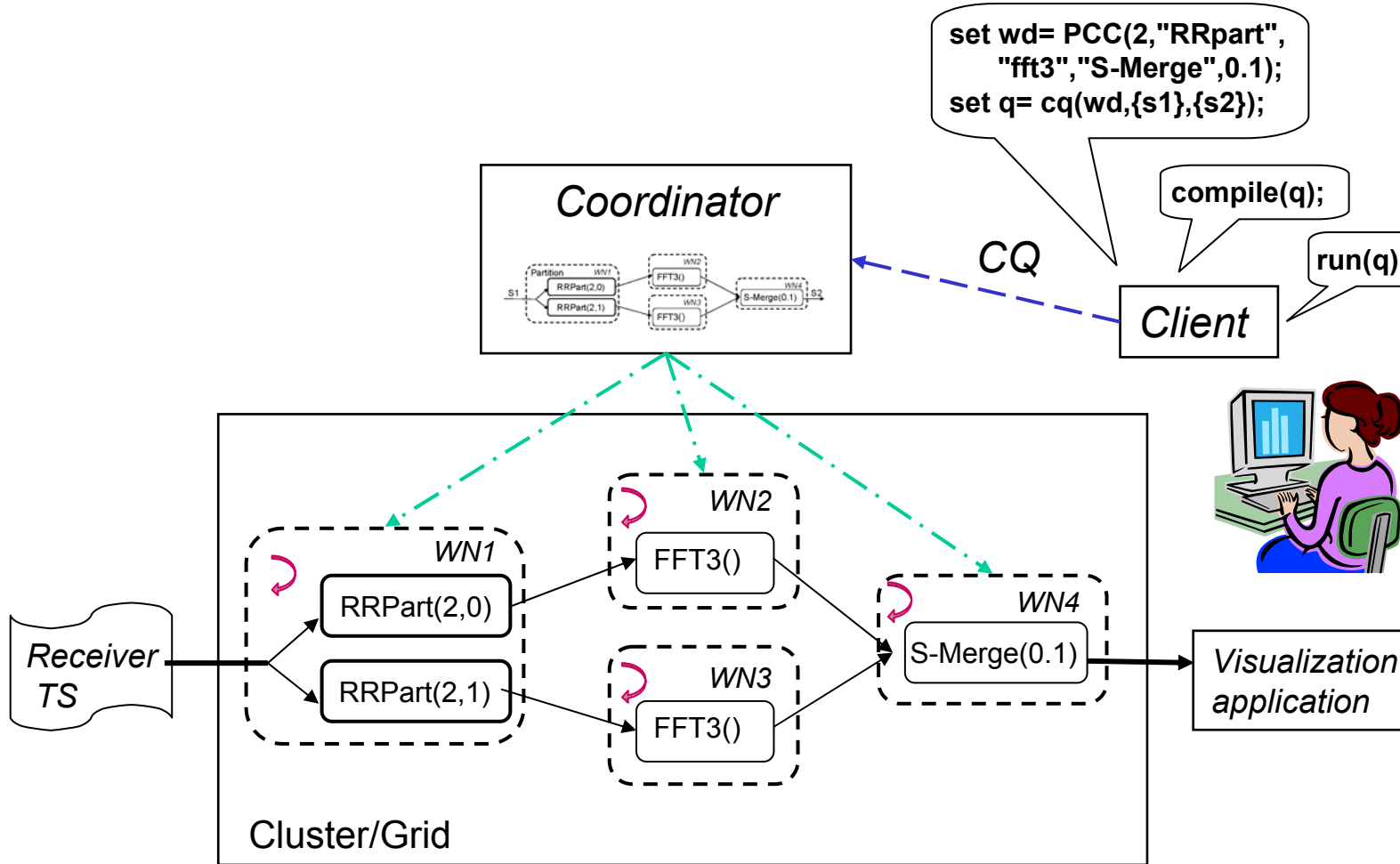


Define computations over windows of data streams through *continuous queries*

- A stream data manager for *very high volume scientific streams* of events
- Main memory stream query engine
- User Defined Functions called in CQs allows *customized computations*
- *Distributed* architecture
- **Data Flow Distribution Templates** allow customized *scalable distributed computation models*



Distributed DSMS Scenario



Legend:

- Client request
- Control flow
- Data flow



Continuous queries

- CQs return event streams
 - C.f. stream of notifications
- CQ events (elements) often *time stamped*
 - C.f. temporal events $e(t)$
- CQs often aggregate over *stream windows*
 - C.f. $e(t,X)$ where X is vector or bag
- Various kinds of joins over streams
 - C.f. Theta joins, temporal joins
- CQs are turned on and run until stop condition true
 - C.f. (de)activate ECA-rules



Data Flow Distribution Templates

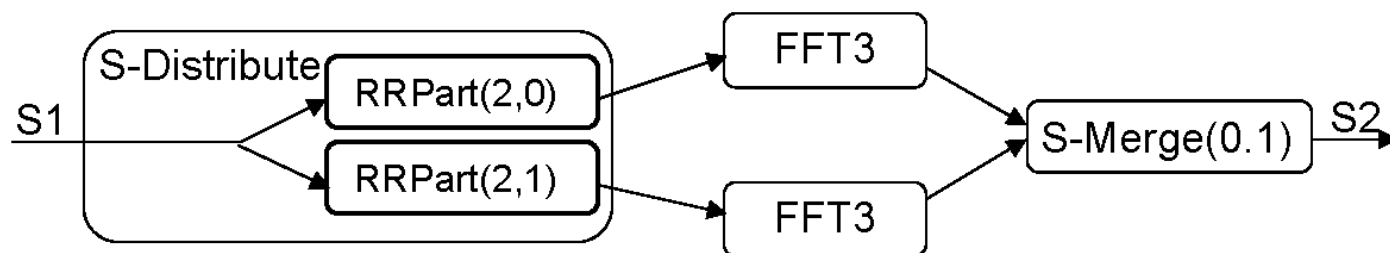
- High level primitives to define different *parallel execution strategies* for different kinds of CQs
- DFDTs can be *application independent* and applicable on any window operation:
 - e.g. **Window Distribute** does partitioning (e.g. Round Robin) as in distributed databases.
- DFDTs can be *application specific*:
 - E.g. **Window Split** does customized splitting of large windows to smaller ones e.g. RadixFFT for distributed FFT
- DFDTs can be *generic* and *combine* other DFDTs
 - E.g. **PCC (Partition-Compute-Combine)** specifies general lattice-shaped partitioned parallelisms



Window Distribute

- Distributes *several* logical windows among different partitions
- Parameterized on partitioning function

```
set wd= PCC(2,"S-Distribute","RRpart",  
         "fft3","S-Merge",0.1);
```

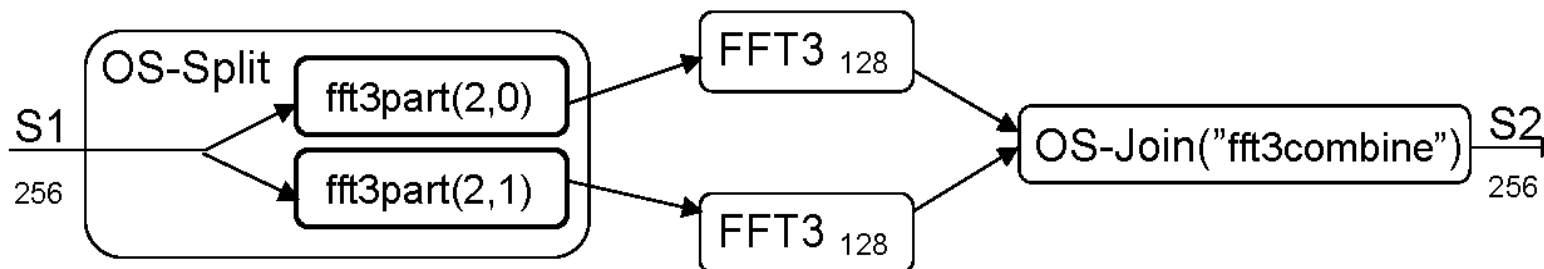




Window Split

- *Application-dependent* parameters for window splitting and combining
- Splits a *single* logical window into sub-windows

```
set ws= PCC(2,"OS-Split","fft3part",  
         "fft3","OS-Join","fft3combine");
```

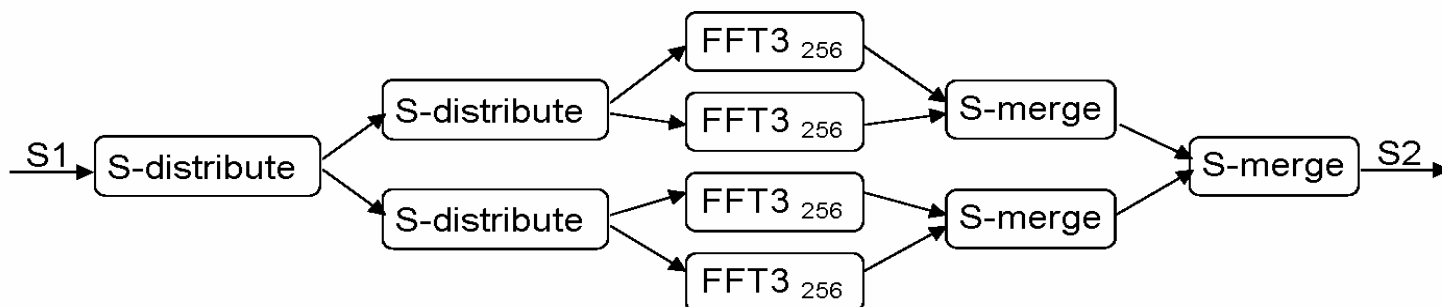




Composed data flow templates

Tree-structured partitioning and combining :

```
set wd-tree =  
    PCC(2, "S-Distribute", "RRpart",  
        "PCC", (2, "S-Distribute",  
                  "RRpart", "fft3",  
                  "S-Merge", 0.1),  
        "S-Merge", 0.1);
```

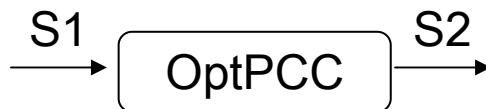




Optimizing template

Invokes enumeration of template generation and profiling (Thesis):

```
set q = opt("PCC", "fft3");
```





E.Zeitler and T.Risch: Processing high-volume stream queries on a supercomputer, *ICDE Ph.D. Workshop 2006*, Atlanta, GA, April 2006

E.Zeitler and T.Risch: Using stream queries to measure communication performance of a parallel computing environment. Proc. First International Workshop on Distributed Event Processing, Systems and Applications (DEPSA), Toronto, Canada, June 29, 2007.



SCSQ, Super Computer Stream Query processor



Erik Zeitler

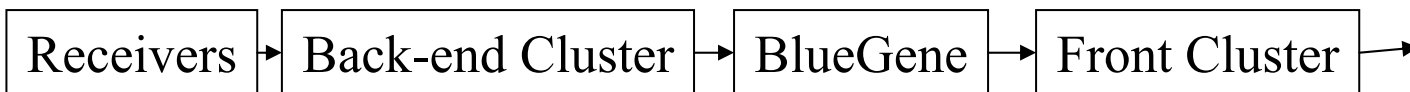
Purpose:

Scalable search and processing of very high-volume data produced by scientific instruments on *massively parallel computers*

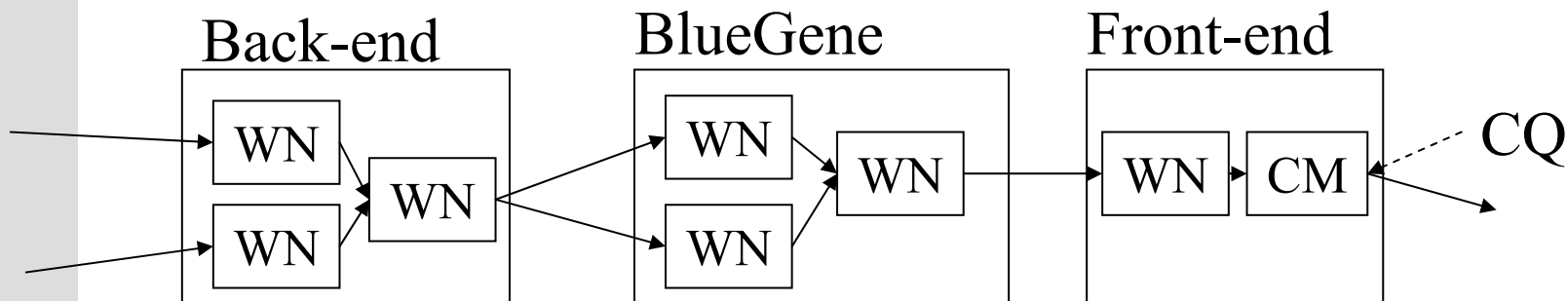
- Distributed DSMS for extremely high-volume data streams
- Dynamic specification of parallel continuous queries
- Utilize state-of-the art hardware (IBM BlueGene for LOFAR)
- Optimize CQs across *several different* clusters



LOFAR stream flow:



SCSQ processors/processes:





SCSQ

- Stream query language SCSQL allowing specification of dynamic parallel CQs
 - Not separate static template language as in GSDM
 - Dynamic parallelization during query execution
- Runs on many platforms
 - IBM BlueGene
 - Linux cluster (IBM, Dell)
 - Windows
- Allows distribution and parallelization of CQs across heterogeneous platforms
- Presented at DEPSA 2007 in Toronto