Stream and Complex Event Processing

Benchmarking Information Flow Processing Systems

G. Cugola E. Della Valle A. Margara

Politecnico di Milano
cugola@elet.polimi.it
dellavalle@elet.polimi.it

Vrije Universiteit Amsterdam
a.margara@vu.nl

On Empirical evaluation of systems 1/2

- In 1995, "experimental evaluation in Com-
- puter Science: a quantitative study” by Tichy and collaborators [1] observed that
  - Computer scientist publish few paper with experimentally validated results
    - 40% no validation (12-15% in other fields)
  - In the papers that contains some evaluation, the fraction of the paper devoted to present it is small
    - 30% (70% in other fields)
- The lack of evaluation was a serious weakness in computer science in 1995

On Empirical evaluation of systems 2/2

- In 2009, "Empirical evaluation in Computer Science research published by ACM" by Wainer and collaborators [2], repeated the study in [1] for the paper published in 2005 and observed that
  - 70% of the paper are about design and modeling
    - 4% theory, 17% empirical, 4.7% hypothesis testing, and 3.4% other
  - Within the design and modeling class, 33% of the papers have no evaluation
  - Same results as 1995
- The lack of evaluation is still a serious weakness in computer science

Benchmarking databases

- The bible:
  - Database and Transaction Processing Performance Handbook by Jim Gray [3]
- The question:
  - What system does the job with the lowest cost-of-ownership?
- The solution
  1. **Define** a **benchmark** (or workload)
  2. **run** on several different systems
  3. **measure**
     - the **performance**: a throughput metric (work/second)
     - the **price**: a five-year cost-of-ownership metric

The Need for Domain-Specific Benchmarks

- Generic benchmarks give some sense of the relative performance and price/performance of a system
- No single metric can measure the performance of computer systems on all applications.
- Domain-specific benchmarks are a response to diversity of computer system use.
  - Each benchmark specifies a synthetic workload characterizing typical applications in that problem domain.
  - The performance of this workload on various computer systems then gives a rough estimate of their relative performance on that problem domain.

The Key Criteria For a Domain-Specific Benchmark

- Relevant
  - measures performance and price/performance of systems when performing typical operations within the problem domain
- Portable
  - easy to implement on many different systems
- Scalable
  - applies to small and large computer systems
- Simple
  - understandable, otherwise it will lack credibility.

Dimension to be benchmarked

- Complex Queries
- Complex Transactions
- Utility Operations
  - e.g., building indexes, altering tables
- Mixed Workloads

Data base benchmarks in [3] 1/2

- TPC BM™ A/B/C
  - Online Transaction Processing
    - http://research.microsoft.com/en-us/um/people/gray/BenchmarkHandbook/chapter2.pdf
- Wisconsin
  - Relational Queries
    - http://research.microsoft.com/en-us/um/people/gray/BenchmarkHandbook/chapter4.pdf
- AS^3AP
  - Mixed Workload of Transactions, Relational Queries, and Utility Functions
    - http://research.microsoft.com/en-us/um/people/gray/BenchmarkHandbook/chapter5.pdf
- Set Query Benchmark:
  - Complex Queries and Reporting
    - http://research.microsoft.com/en-us/um/people/gray/BenchmarkHandbook/chapter6.pdf
- Engineering Database Benchmark:
  - Engineering Workstation-Server
    - http://research.microsoft.com/en-us/um/people/gray/BenchmarkHandbook/chapter7.pdf

Benchmarking databases

Data base benchmarks in [3] 2/2

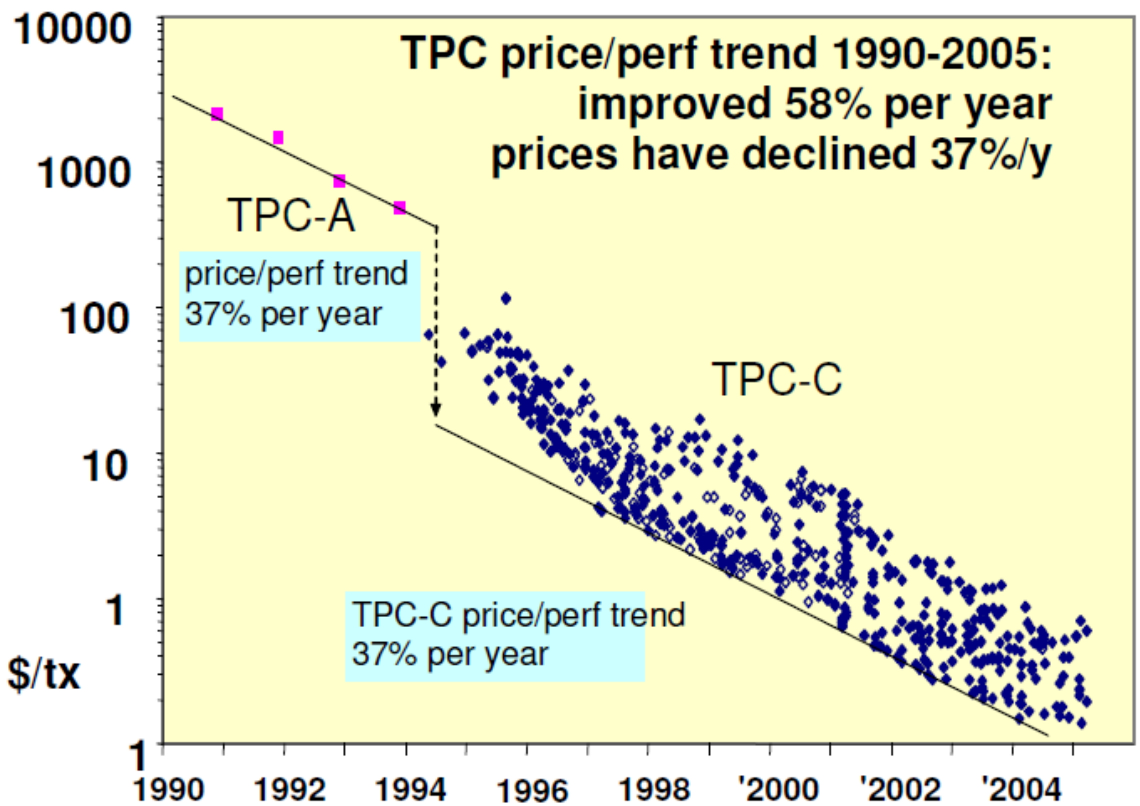
Dimension	Complex Queries	Complex Transactions	Utility Operations	Mixed Workloads
TPC BM™ A/B/C	x	x		
Wisconsin	x			
AS^{3}AP	x	Simple only	x	x
Set Query Benchmark	x			
Engineering Database Benchmark	simple	simple	x	

How to Use Those Benchmarks

- compare different software products on one machine
- compare different software products on one machine
- compare different machines in a compatible family
- compare different releases of a product on one machine

Benefits of benchmarking

- make competing products comparable
- accelerate progress, make technology viable



What about DSMS/CEP/SR? [4]

- Different properties
- New challenges
- New key performance indicators
- New stress tests

Different properties

- Time Model
  - Implicit vs. explicit
- Time Semantics
  - Punctual vs. interval
- Query Model
  - Stream analysis vs. event pattern matching
- Quality of Service
  - Best effort vs. guaranteed QoS
- Distribution
- Background Data support
- Inference Support

What about DSMS/CEP/SR?

New challenges

- Time Modeling
- Querying
- Managing Bursts
- Managing Background Data
- Inference Expressivity

New challenges - Time Modeling

- Choosing a specific model –and a corresponding semantics– for representing time can significantly impact the performance of the system

New challenges - Querying

- Choosing an appropriate strategy for storing, accessing, and discarding partial results
- Choosing operators that determine the scope of processing
- Choosing mechanism for triggering queries and the management of multiple queries

New challenges - Managing Bursts

- Scarify completeness and correctness vs. scale-out and scale-in back

What about DSMS/CEP/SR?

New challenges - Managing Background Data

- If the background knowledge is large or pulling it on demand is time consuming it can become a bottleneck for real-time processing
- Large background data may be difficult to distribute

New challenges – Inference Expressivity

- Choosing the right expressivity
  - ++expressivity → ++processing time
- Temporal reasoning vs. built-in time-management

What about DSMS/CEP/SR?

Properties and challenges

	Time Modeling	Querying	Managing Bursts	Managing Background Data	Inference Expressivity
Time Model	x	x			x
Time Semantics	x	x			x
Query Model		x		x	x
Quality of Service	x	x	x		x
Distribution		x	x	x	x
Background Data support		x		x	x
Inference Support		x			x

Key performance indicators (KPIs)

- DSMS/CEP/SR are *reactive systems*
  - Answer must arrive within a given time
  - Answers received after that time are useless
- KPIs
  - Response time
    - Average/x^{th} Percentile/Minimum/Maximum
  - Maximum input throughput
  - Time to accuracy
    - hopefully equal to response time
  - Time to completion
    - not necessarily equal to response time
  - Minimize Resource utilization
    - RAM, bandwidth

What to benchmark for

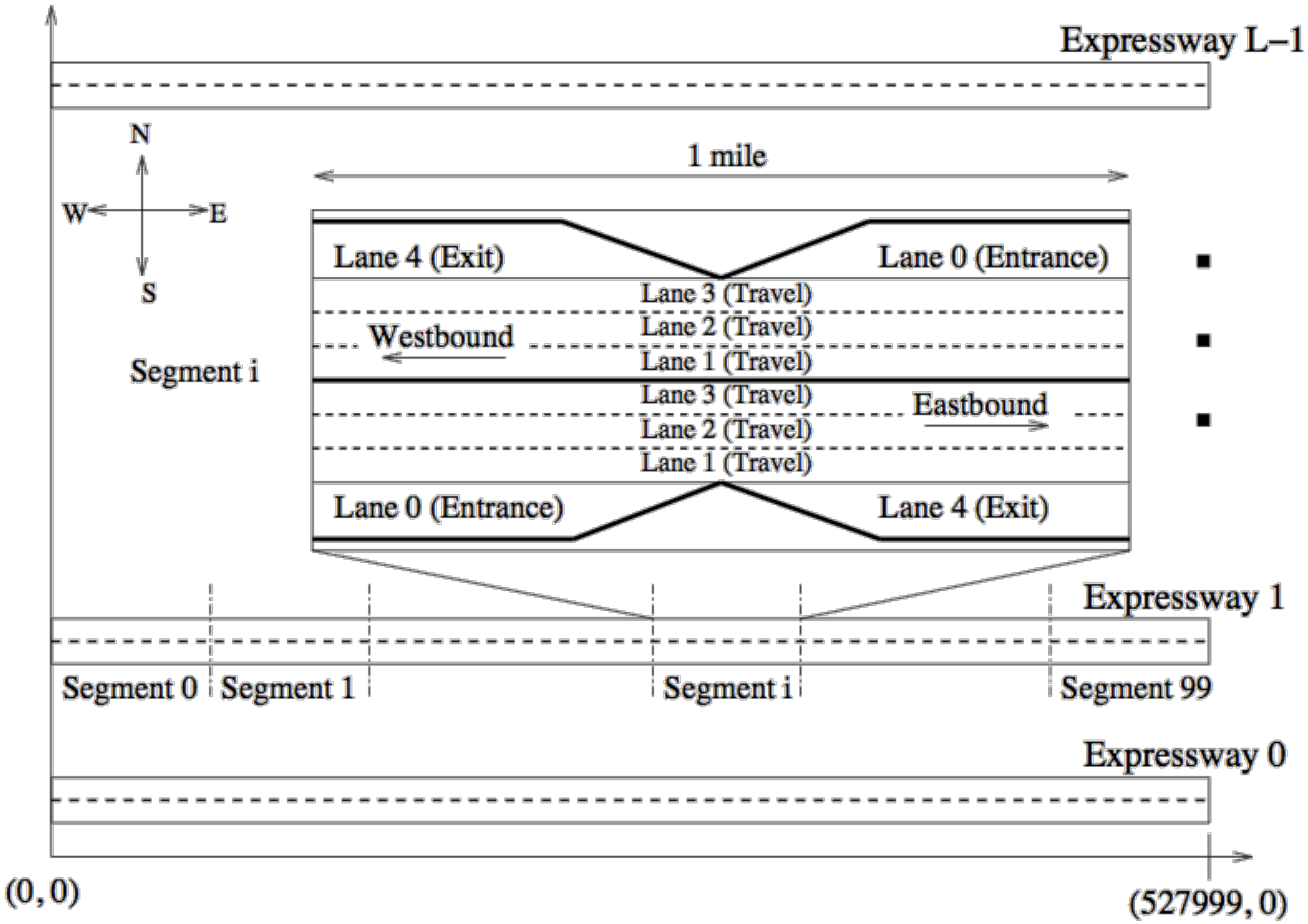
- Load Balancing
- Filter, Joins (and Inference) on Flow Data Only
- Filter, Joins and Inference in Flow and Background Data
- Aggregates
- Unexpected Data (out of order, and noisy)
- Schema size and expressiveness
- Changes in Background-Data

Existing benchmarks

- DSMS
  - Linear Road Benchmark [5]
    - http://www.cs.brandeis.edu/~linearroad/
- CEP
  - Fast Flower Delivery [6]
    - http://www.ep-ts.com/content/view/80
- Stream Reasoners (I'd better to say RDF stream processor)
  - SR-Bench [7]
    - http://www.w3.org/wiki/SRBench
  - LS-Bench [8]
    - http://code.google.com/p/lsbench/

Linear Road Benchmark – The idea

- Simulates an unban highway system that uses ‘variable tolling’ (i.e, congestion-based pricing).

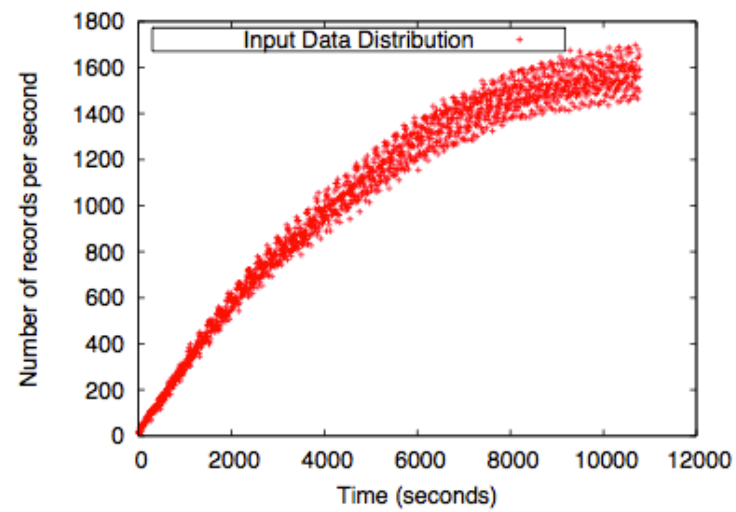


Linear Road Benchmark – The Challenges

- Semantically Valid Input
  - Use a simulator: MIT Traffic Simulator
- Many Correct Results
  - continuous queries results may depend upon evolving historical state or the arrival order tuples on a stream, and therefore several different results for the same query may be “correct”
- No Query Language
  - Queries are language-agnostic, yet have a clear semantics

Linear Road Benchmark – The data

- A 3-hour, single expressway worth of input data consists of
  - 12 million position reports
  - 67000 account balance
  - 14000 daily expenditure query requests.
- As the simulation time progresses, the input data exhibits a monotonically increasing distribution with time
  - from 15 records per second to 1700 records per second.



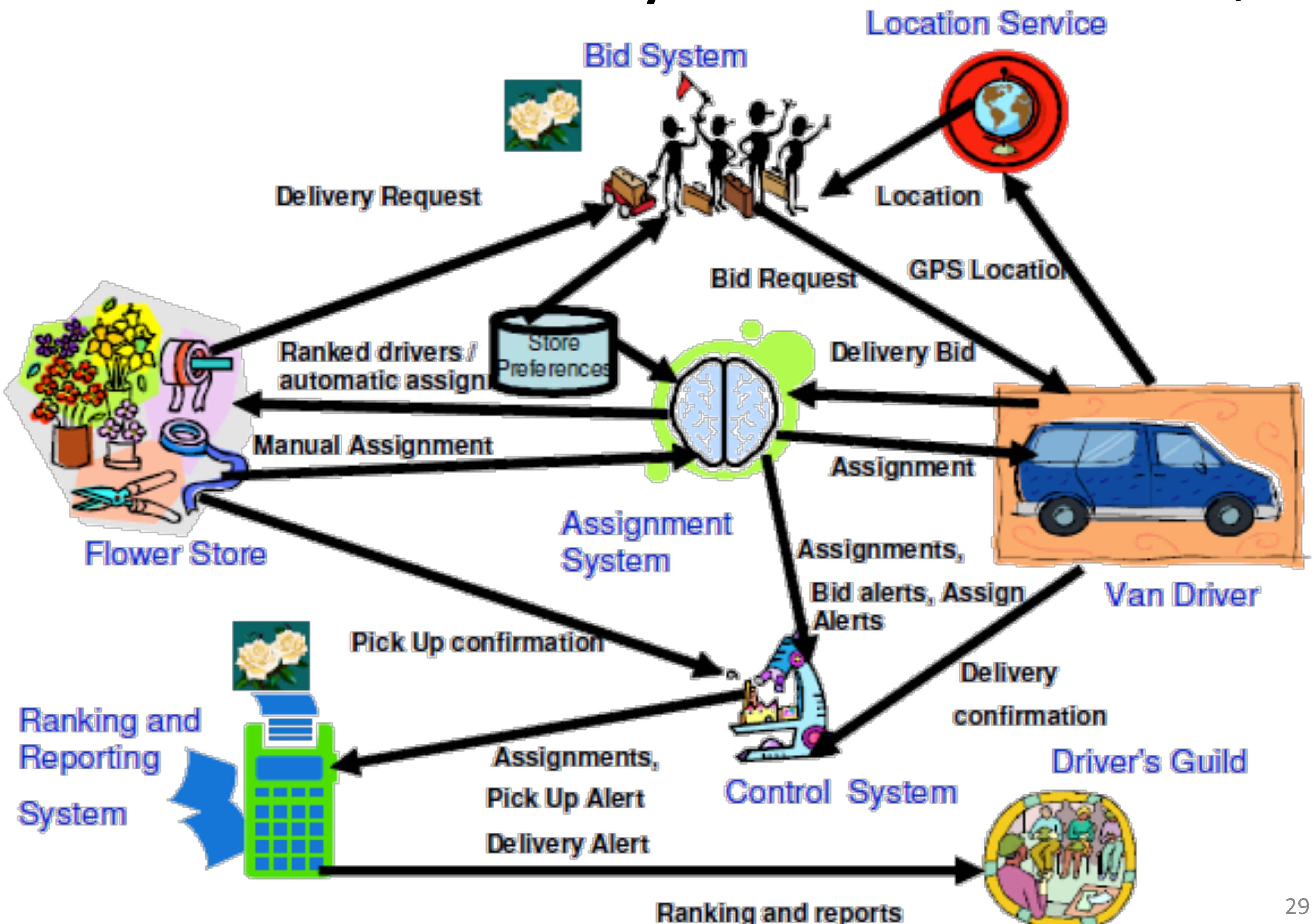
Linear Road Benchmark – The queries

- Two continuous queries
  - Calculates a segment toll every time a vehicle enters the segment.
  - Detects and reports accidents and adjusts tolls accordingly.
- Three Historical queries
  - Request an account balance
  - Day's total expenditure for a given vehicle
  - Prediction of travel time between two segments using historical data
- KPI
  - Each of these queries must be answered with a specified accuracy and within a specified response time.

Fast Flower Delivery – The idea 1/2

- The flower stores association in a large city has established an agreement with local independent van drivers to deliver flowers from the city's flower stores to their destinations.
- The CEP-centric system is used to support this business need.
- No KPIs
- More a use case than a benchmark

Fast Flower Delivery – The idea 2/2



Fast Flower Delivery – The phases 1/2

1. Bid phase
   - When a store gets a flower delivery order, it creates a request, which is broadcasted to relevant drivers within a certain distance from the store, with the time for pick up (typically now) and the required delivery time if it is an urgent delivery.
2. Assignment phase
   - A driver is then assigned and the customer is notified that a delivery has been scheduled.
   - Assignment can be both automatic (see phase) or manually performed by the store

Fast Flower Delivery – The phases 2/2

3. Delivery process
   - The driver picks up the delivery and delivers it, and then person receiving the flowers confirms the delivery time by signing for it on the driver's mobile device.
4. Ranking Evaluation
   - The system maintains a ranking of each individual driver based on his or her ability to deliver flowers on time.
5. Activity monitoring
   - Creates reports on driver's performances

Fast Flower Delivery – Additional info

- Each store has a profile that can include a constraint on the ranking of its drivers, for example a store can require its driver to have a ranking greater than 10. The profile also indicates whether the store wants the system to assign drivers automatically, or whether it wants to receive several applications and then make its own choice.

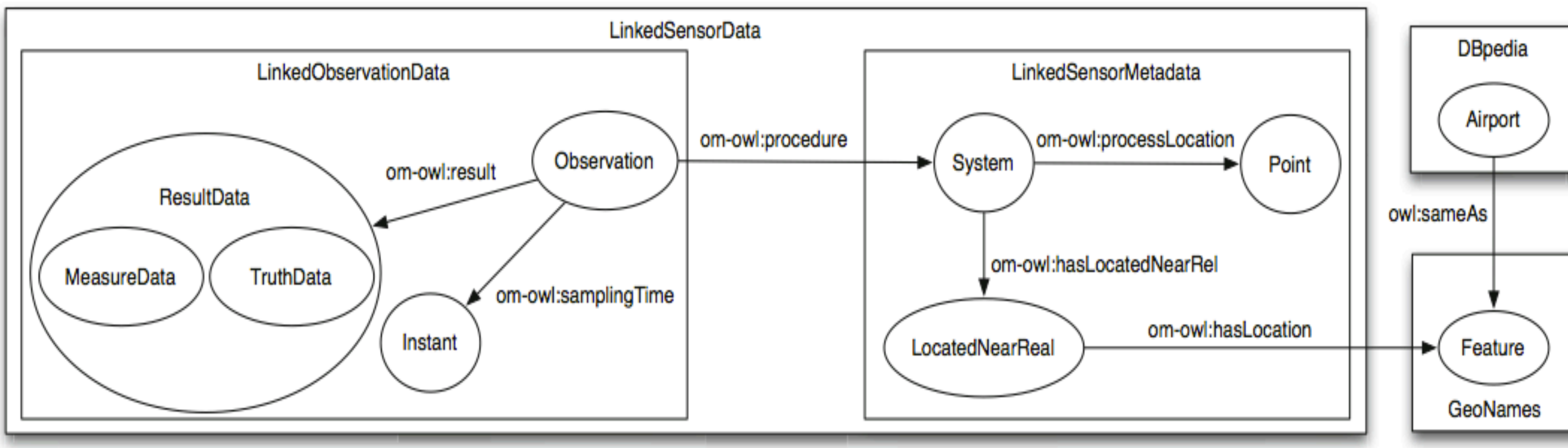
SRbench - Challenges

- Proper benchmark dataset
  - use real-world datasets from LOD
    - LinkedSensorData
- No standard query language:
  - natural language query definition and
  - three implementations
    - SPARQLStream
    - CQELS
    - C-SPARQL

Existing benchmarks

SRbench - Datasets

- Use case: weather information application
- Data model



- Data:
  - 10.000 weather station
  - 5 sensor in each: temperature, visibil- ity, precipitation, pressure, wind speed and humidity
  - Millions of sensor observations

Existing benchmarks

SRBench - Queries

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17
1 patter matching	A	A,F,O	A	A,F	A	A,F,U	A	A	A	A	A,F	A,F,U	A,F	A,F,U	A,F	A,F	A,F
2 solution modifier	P,D	P,D	P	P	P	P	P,D	P	P	P,D	P,D	P	P	P,D	P	P	P
3 query form																	
4 SPARQL 1.1		F,P	A	A,E,M,F	A,S		N	A,E,M	A,E,M		A,S,M,F	A,S,E,M,F,P	A,E,M,F,P	F,P	A,E,M,P	P	P
5 reasoing			R												C	A	C
6 CQL feature	T	T	T	T	T	T	T,	T	T	T	T	T	T	T	T		
7 data access	O	O	O	O	O	O	O	O, S	O,S	O,S	O,S	O,S,F	O,S,G	O,S,G	O,S,D	O,S,G,D	S

Table 2. Addressed features per query. Operators are abbreviated in per row unique capital letters, defined as: 1. **A**nd, **F**ilter, **U**nion, **O**ptional; 2. **P**rojection, **D**istinct, **L**imit; 3. **S**elect, **C**onstruct, **A**sk; 4. **A**ggregate, **S**ubquery, **N**egation, **E**xpr in SELECT, assign**M**ent, **F**unctions&operators, **P**roperty path; 5. sub**C**lassOf, subp**R**opertyOf, owl:same**A**s; 6. **T**ime-based window, tu**P**le-based window, **I**stream, **D**stream, **R**stream; 7. Linked**O**bservationData, Linked**S**ensorMetadata, **G**eoNames, **D**bpedia.

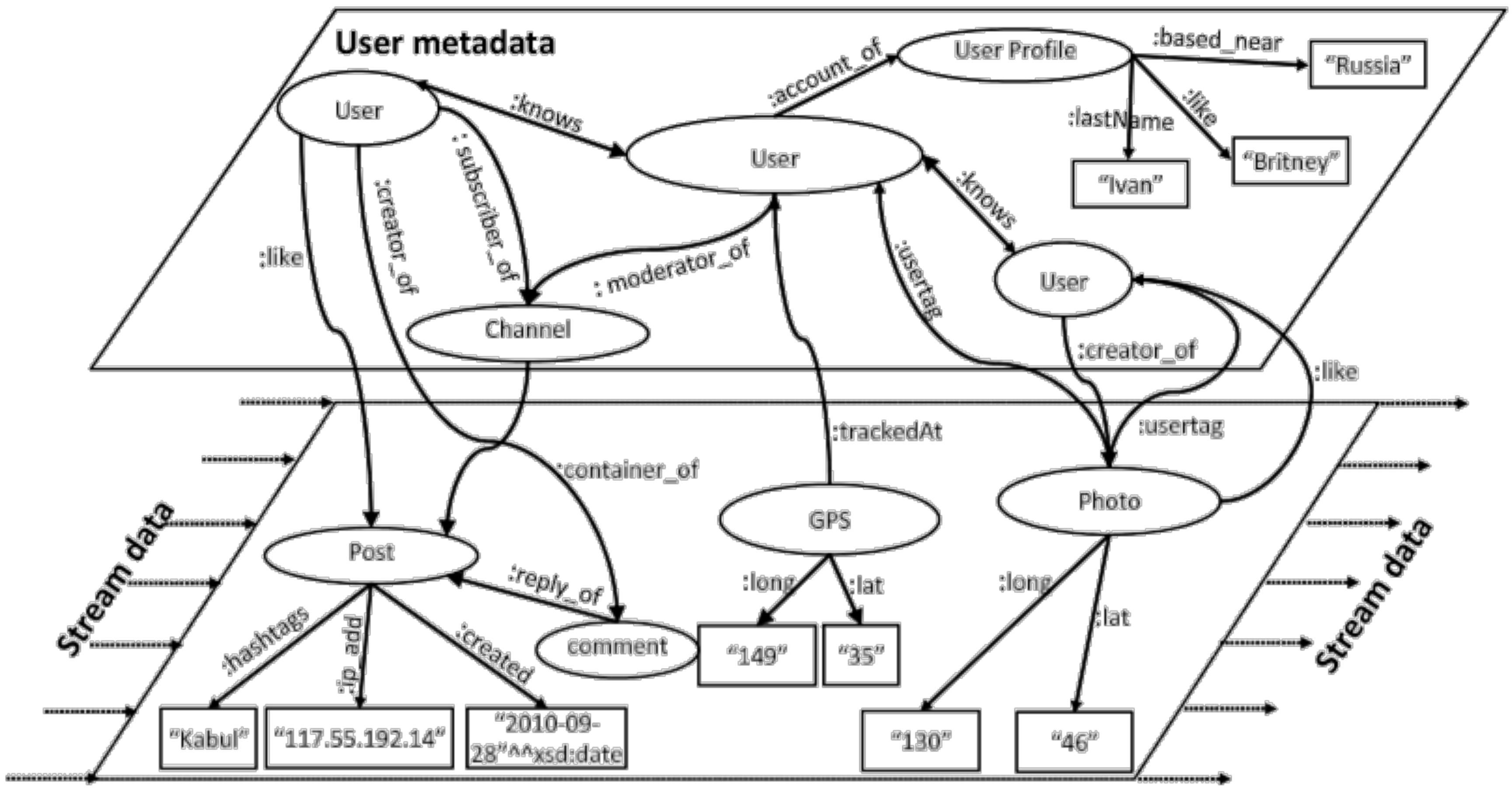
SRbench - KPI

- Feature coverage
- Correctness
  - In case of approximate answers: precision and recall
- Throughput
- Scalability
- Response Time

Existing benchmarks

LS-Bench – Use case

- Social Network + Social Streams



LS-Bench – Data set

- Synthetic data generation
  - http://lsbench.googlecode.com/files/sibStream0617.tar
- It resembles reality
  - In the social network form and dynamics
  - In the skewed distribution of posts/comments
  - In GPS tracks
- Customizable
  - Duration
  - Maximum number of posts/comments/photos for each user per week
  - Correlation probabilities

LS-Bench – Queries

	Patterns covered							S	N_P	N_S
	F	J	A	E	N	U	T			
Q_1	√								1	1
Q_2		√						√	2	1
Q_3		√						√	3	1
Q_4	√	√							4	1
Q_5		√						√	3	2
Q_6		√						√	4	2
Q_7	√	√						√	7	2
Q_8		√			√				3	2
Q_9	√	√				√		√	8	4
Q_{10}			√						1	1
Q_{11}		√	√	√					2	2
Q_{12}			√				√		1	1

Legend

F: Filter
J: join
A: aggregation
E: nested query
N: negation
U: union
T: top-k
S: use static data
N_P: number of patterns
N_S: number of streams

LS-Bench – KPI

- Feature coverage
- Correctness
  - by comparison of results of different implementations ☹
- Performance: input throughput

Evaluation of existing benchmarks [4]

Criteria	Linear Road Benchmark	Fast Flower Delivery	SR-bench	LS-Bench
load balancing	yes	no	no	yes
Filter, Joins (and Inference) on Flow Data Only	yes (no)	yes (no)	yes (yes)	yes (no)
Filter, Joins (and Inference) in Flow and Background Data	yes (no)	yes (no)	yes (no)	yes (no)
Aggregates	yes	yes	yes	yes
Unexpected Data	no	no	no	no
Schema size expressiveness	yes(no)	yes(no)	yes(no)	yes(no)
Changes in Background-Data	yes	yes	no	yes

References

1. Walter F. Tichy, Paul Lukowicz, Lutz Prechelt, Ernst A. Heinz: Experimental evaluation in computer science: A quantitative study. Journal of Systems and Software 28(1): 9-18 (1995)
2. Jacques Wainer, Claudia Galindo Novoa Barsottini, Danilo Lacerda, Leandro Rodrigues Magalhães de Marco: Empirical evaluation in Computer Science research published by ACM. Information & Software Technology 51(6): 1081-1085 (2009)
3. Gray, J.: The Benchmark Handbook for Database and Transaction Systems. Morgan Kaufmann, 2nd edn. (1993)
4. Thomas Scharrenbach, Jacopo Urbani, Alessandro Margara, Emanuele Della Valle, Abraham Bernstein: Seven Commandments for Benchmarking Semantic Flow Processing Systems. Extended Semantic Web Conference 2013. TO APPEAR
5. Arvind Arasu, Mitch Cherniack, Eduardo F. Galvez, David Maier, Anurag Maskey, Esther Ryvkina, Michael Stonebraker, Richard Tibbetts: Linear Road: A Stream Data Management Benchmark. VLDB 2004: 480-491
6. Etzion, O., Niblett, P.: Event Processing In Action. Manning Publications Co., Greenwich, CT, USA (2010)
7. Ying Zhang, Minh-Duc Pham, Óscar Corcho, Jean-Paul Calbimonte: SRBench: A Streaming RDF/SPARQL Benchmark. International Semantic Web Conference (1) 2012: 641-657
8. Danh Le Phuoc, Minh Dao-Tran, Minh-Duc Pham, Peter A. Boncz, Thomas Eiter, Michael Fink: Linked Stream Data Processing Engines: Facts and Figures. International Semantic Web Conference (2) 2012: 300-312