University of Zurich[UZH]

Department of Informatics

VU VRIJE UNIVERSITEIT AMSTERDAM

Seven Commandments for Benchmarking Semantic Flow Processing Systems

Thomas Scharrenbach,

Jacopo Urbani,

Alessandro Margara,

Emanuele Della Valle, and

Abraham Bernstein

SEVENTH FRAMEWORK PROGRAMME

vista TV

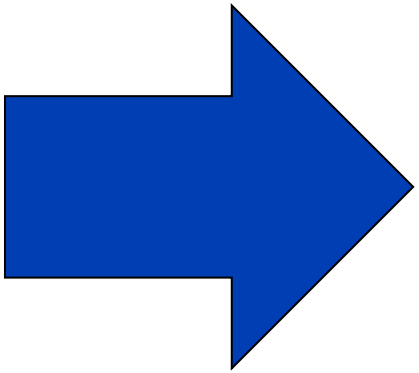
COMMIT/

How close are
we from the
Perfect Benchmark
for Semantic Flow
Processing Systems?

The nature of flows requires *a paradigmatic change**

Persistent data

- to be stored and queried **on demand**
- a.k.a. ***one time semantics***



Transient data

- to be **consumed on the fly** by continuous queries
- a.k.a. ***continuous semantics***

*

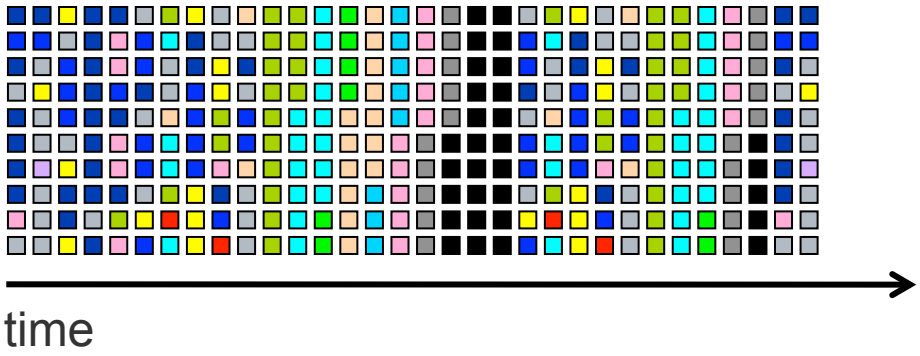
*This paradigmatic change first arose in DB community

Transient Data - Flow Processing Systems

What are data flows?

Formally:

- *Data flows are unbounded sequences of time-varying data elements*

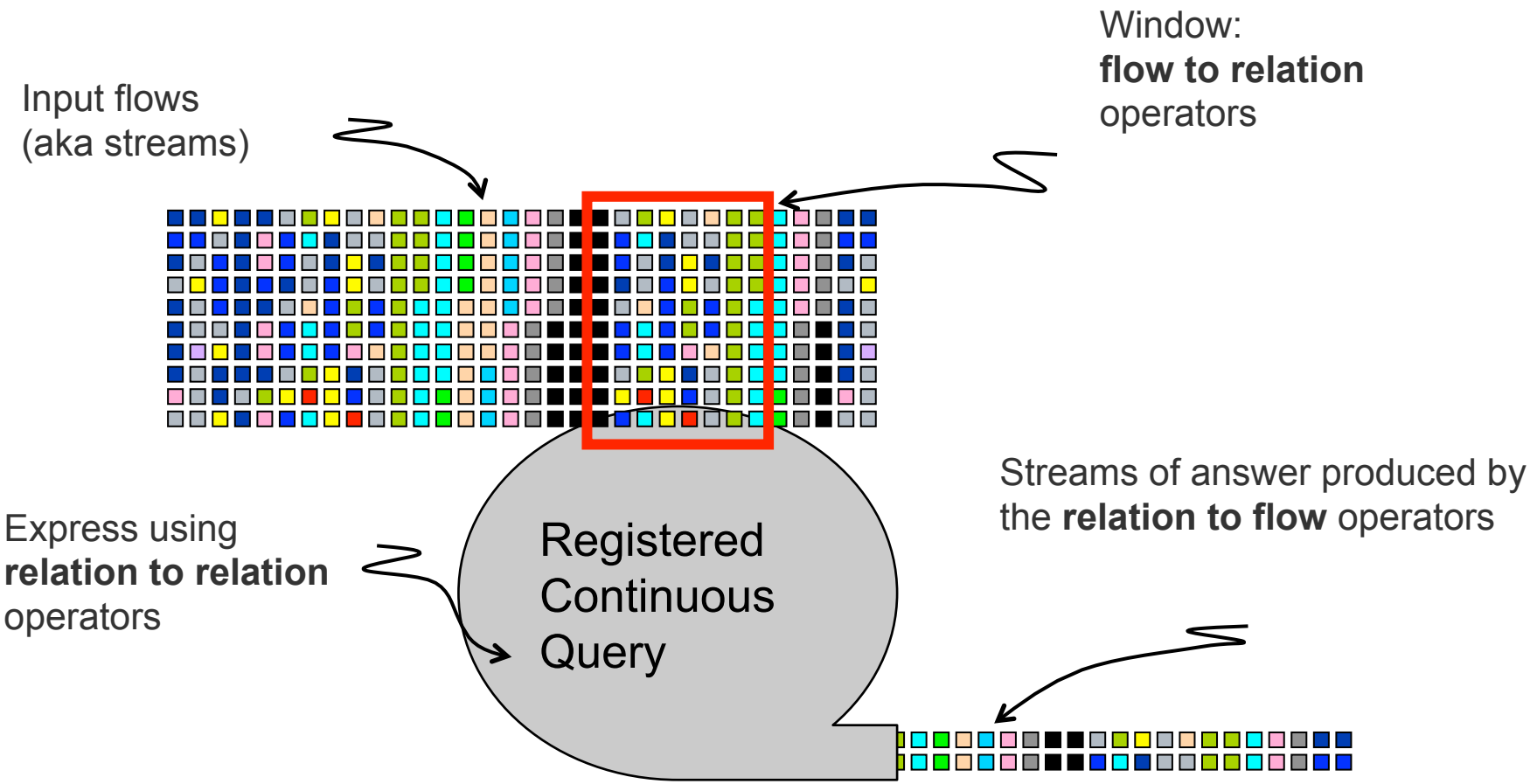


Less formally:

- an (almost) “continuous” ***flow of information***
- with the recent information being more relevant as it describes the ***current state of a dynamic system***

Flow Processing Systems

Continuous queries registered over streams that, in most of the cases, are observed trough **windows**



Great, but what about **semantics**?

Semantics – the missing bit for Information Flow Processing

Information Flow Processing		Semantic Web
✓	Continuous semantics	✗
✓	Scalable processing	✓
✓	Real-time systems	✗
✗	Powerful query languages	✓
✗	Rich ontology languages	✓

Semantic Days 2012 - Emanuele Della Valle - http://streamreasoning.org

Semantics – the missing bit for Information Flow Processing

Semantic Flow Processing	
Continuous semantics	✓
Scalable processing	✓
Real-time systems	✓
Powerful query languages	✓
Rich ontology languages	✓

Semantic Days 2012 - Emanuele Della Valle - http://streamreasoning.org

From Triples Stores to Flows of RDF

1. Extend RDF data model with the notion of
RDF Stream

$$\ldots$$
$$\langle\ s_i\ \ p_i\ \ o_i\ \rangle : [\tau_1]$$
$$\langle\ s_{i+1}\ \ p_{i+1}\ \ o_{i+1}\ \rangle : [\tau_{1+1}]$$
$$\ldots$$

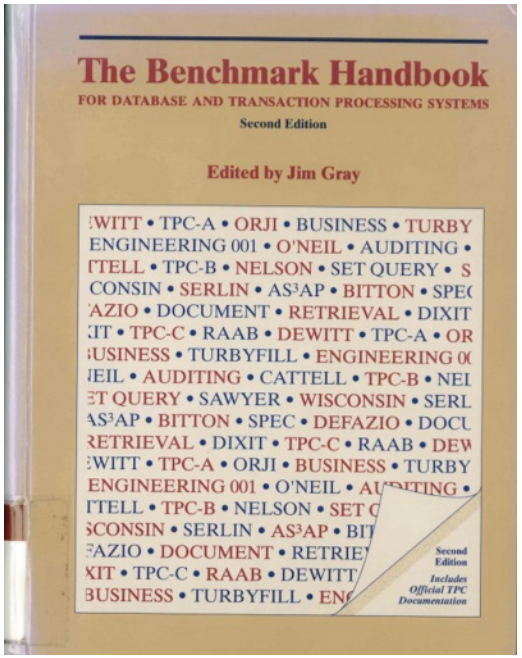
Timestamps are non-decreasing to allow for expressing contemporaneity

2. Extend SPARQL
to express and process **continuous queries**

Existing languages/engines

- CQELS
- $\text{SPARQL}_{\text{STREAM}}$
- C-SPARQL
- EP-SPARQL

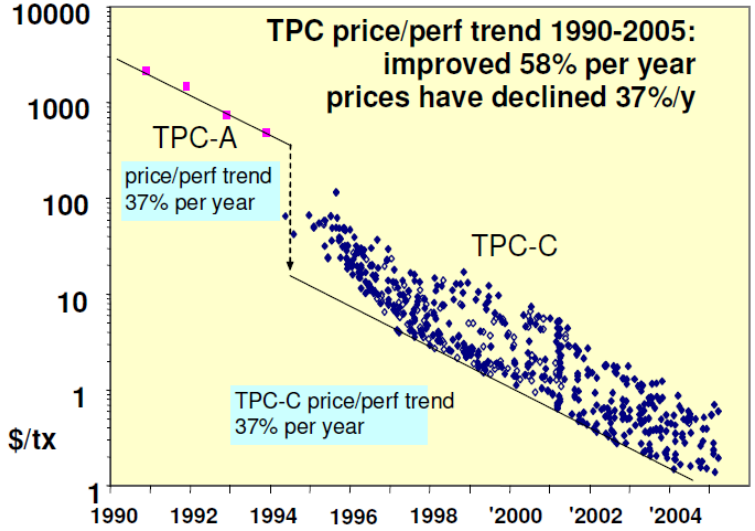
Benchmarking



What system does the job with the ***lowest cost-of-ownership***?

The solution

- Define a benchmark (or workload)
- run on several different systems
- KPI:
  - throughput metric (work/sec)
  - five-year cost-of-ownership metric



Gray & Levine 2012

Why benchmarking?

- make competing products comparable
- accelerate progress, make technology viable
- **scientific method**
- **highlight both strong and weak points**

Why Should You Care?

Most papers in Computer Science **describe systems**…

... few papers **systematically evaluate** these systems.

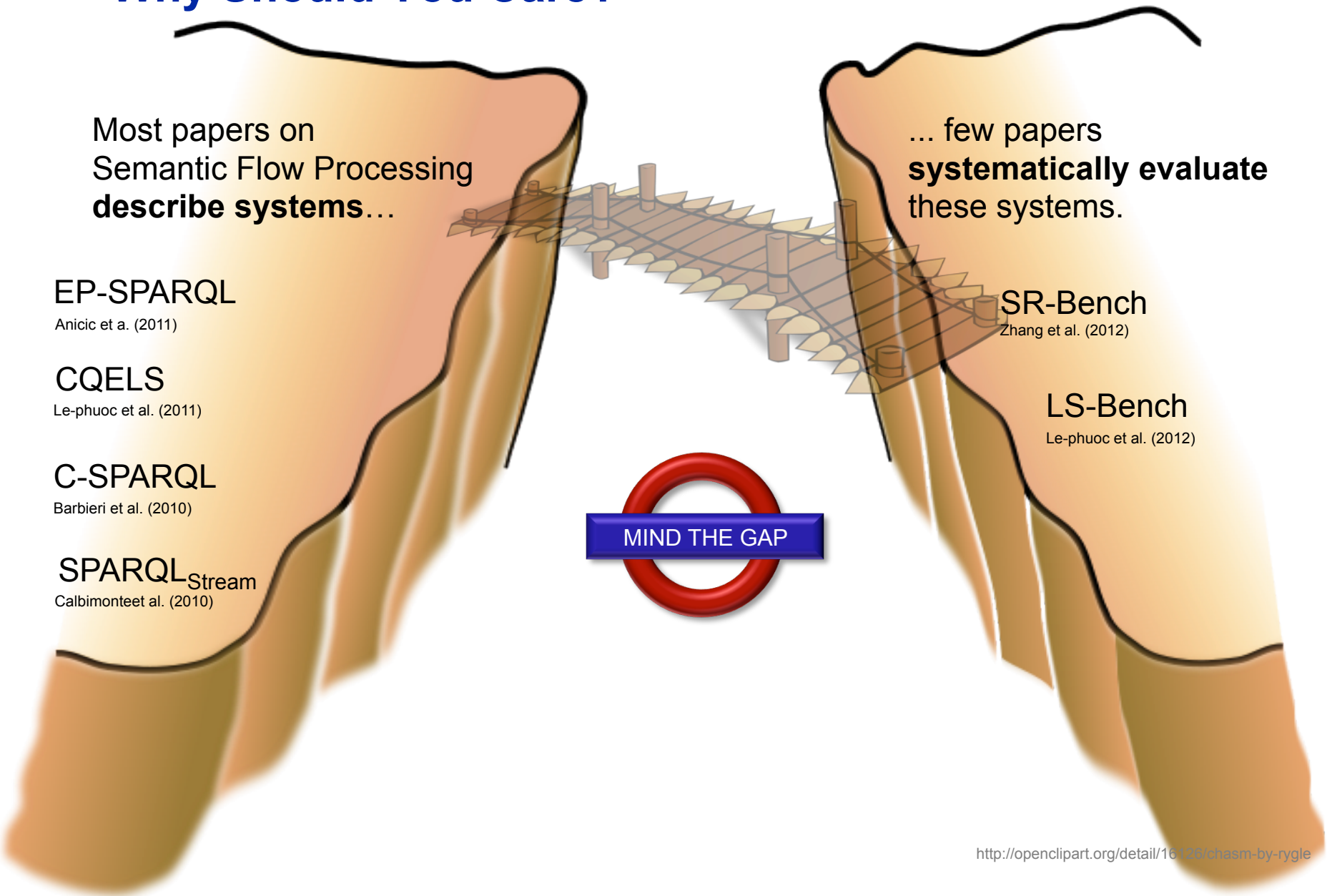
Tichy et al., 1995

Wainer et al., 2009

Computer Science research has not increased significantly its empirical or experimental component

http://openclipart.org/detail/16126/chasm-by-rygle

Why Should You Care?



http://openclipart.org/detail/16126/chasm-by-rygle

Background

Benchmarking RDF stream engines

Relational

Linear Road

Oracle for validation

- Dataset: simulator
- No queries but use-case specification plus validator.
- KPI: feature coverage and correctness

Fast Flower Delivery

- Use-case description with expected results.
- Must-to-implement for commercial CEP systems.

Graph

SRBench

Not verified

- Dataset: LinkedS… meteorological sensor data)
- Queries: 17 continuous queries, some requiring RDFS reasoning
- KPI: feature coverage and correctness

LSBench

Verified comparing the number of results produced by different solutions

- Dataset: synthet… inspired data se…
- Queries: 12 con… multiple stream and static knowledge
- KPI: input throughput and correctness

SFP Systems are Reactive

1. **Answer must arrive within a given time**
2. **Answers received after that time are useless**

Appropriate KPIs:

- **Response time**
  - Average/xth Percentile/Minimum/Maximum
- **Maximum input throughput**
  - The standard KPI
- **Time to accuracy**
  - hopefully equal to response time
- **Time to completion**
  - not necessarily equal to response time
- **Minimize Resource utilization**
  - RAM, bandwidth

The is no benchmark to test them all!

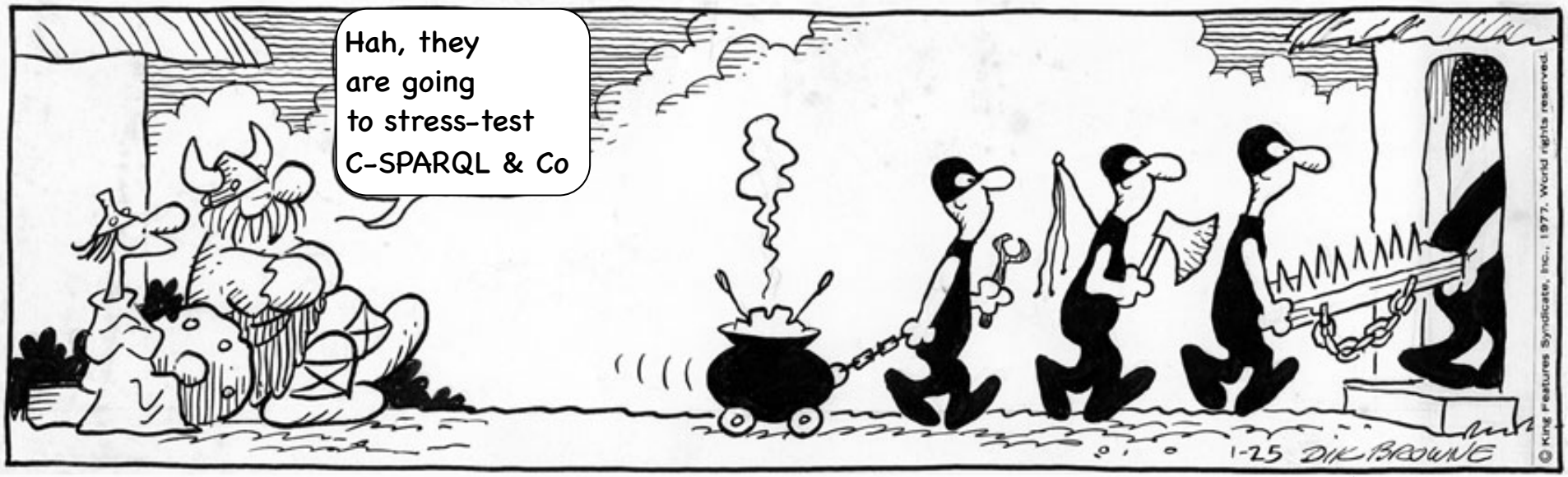
http://en.wikipedia.org/wiki/File:Unico_Anello.png

Does throughput matter without correctness?

Dell'Aglio et al, BerSys 2013

How much does an enhancement in completeness cost?

???



http://greniertv.site88.net/hagar-le-viking-les-comic-strips/

Properties → **C**hallenges → **KPIs** → **S**tress Tests

From **analyzing the key challenges…**

…to a **systematic guideline for assessing the Key Performance Indicators** of SFP systems

How close are
we from the
Perfect Benchmark?

Properties of Semantic Flow Processing Systems

- Support of Background Data
- Inference Support
- Quality of Service (QoS)
- Time Model
- Time Semantics
- Query Model
- Distribution

Challenge C1: Managing Background Data

	LR	FFD	SR-Bench	LS-Bench
S3: Joins and Inference in Flow, BG-Data	✗	✗	✓	✓
S6: Schema	✗	○	✓	○
S7: Changes in Background-Data	✗	✗	○	○

Challenge C2: Expressive Power of Inference

		LR	FFD	SR-Bench	LS-Bench
S1: Load Balancing					
S2: Joins and Inference on Flow Data Only	simple				
	sequential				
	temporal				
S3: Joins and Inference in Flow, BG-Data		✗	✗	✓	✓
S4: Aggregates	shrinking				
	non-shrinking				
S5: Unexpected Data	out-of-order				
	missing				
S6: Schema		✗	○	✓	○
S7: Changes in Background-Data		✗	✗	○	○

Challenge C3: Time Modeling

		LR	FFD	SR-Bench	LS-Bench
S1: Load Balancing					
S2: Joins and Inference on Flow Data Only	simple	✓	✓	✓	✓
	sequential	✓	○	○	○
	temporal	○	✓	○	○
S3: Joins and Inference in Flow, BG-Data					
S4: Aggregates	shrinking	✗	✗	○	○
	non-shrinking	✓	✓	✓	✓
S5: Unexpected Data	out-of-order	○	✗	○	○
	missing	○	○	○	○
S6: Schema					
S7: Changes in Background-Data					

Challenge C4: Querying

		LR	FFD	SR-Bench	LS-Bench
S1: Load Balancing					
S2: Joins and Inference on Flow Data Only	simple	✓	✓	✓	✓
	sequential	✓	○	○	○
	temporal	○	✓	○	○
S3: Joins and Inference in Flow, BG-Data		✗	✗	✓	✓
S4: Aggregates	shrinking	✗	✗	○	○
	non-shrinking	✓	✓	✓	✓
S5: Unexpected Data	out-of-order	○	✗	○	○
	missing	○	○	○	○
S6: Schema					
S7: Changes in Background-Data					

Challenge C5: Managing Bursts

	LR	FFD	SR-Bench	LS-Bench
S1: Load Balancing	○	○	○	○
S2: Joins and Inference on Flow Data Only				
S3: Joins and Inference in Flow, BG-Data				
S4: Aggregates				
S5: Unexpected Data				
S6: Schema				
S7: Changes in Background-Data				

How close are we from the Perfect Benchmark?

		LR	FFD	SR-Bench	LS-Bench
S1: Load Balancing		○	○	○	○
S2: Joins and Inference on Flow Data Only	simple	✓	✓	✓	✓
	sequential	✓	○	○	○
	temporal	○	✓	○	○
S3: Joins and Inference in Flow, BG-Data		✗	✗	✓	✓
S4: Aggregates	shrinking	✗	✗	○	○
	non-shrinking	✓	✓	✓	✓
S5: Unexpected Data	out-of-order	○	✗	○	○
	missing	○	○	○	○
S6: Schema		✗	○	✓	○
S7: Changes in Background-Data		✗	✗	○	○

How close are we from the Perfect Benchmark?

		LR	FFD	SR-Bench	LS-Bench
S1: Load Balancing		○	○	○	○
S2: Joins and Inference on Flow Data Only	simple	✓	✓	✓	✓
	sequential	✓	○	○	○
	temporal	○	✓	○	○
S3: Joins and Inference in Flow, BG-Data		✗	✗	✓	✓
S4: Aggregates	shrinking	✗	✗	○	○
	non-shrinking	✓	✓	✓	✓
S5: Unexpected Data	out-of-order	○	✗	○	○
	missing	○	○	○	○
S6: Schema		✗	○	✓	○
S7: Changes in Background-Data		✗	✗	○	○

How close are
we from ~~the~~
~~Perfect~~ **Benchmarks**?

Appropriate

Limitations

We did not implement a concrete benchmark.

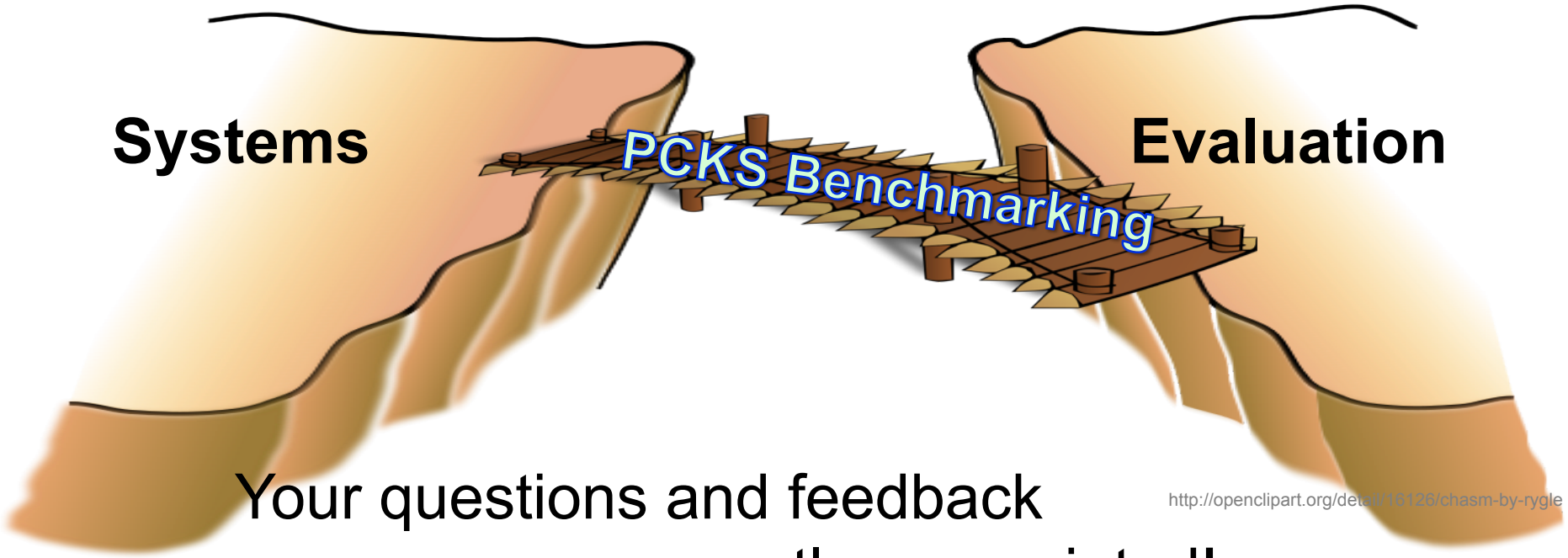
- No bias towards a specific challenge/stress test.
- Distinguish between the abstract definition and its implementation.
- There is no such thing as a universal benchmark.

No standards exist for comparing different SFP systems.

- How can we describe configurations?
- What makes two systems comparable?

Conclusion

1. SFP research needs to become **more empirical**.
2. There are more KPIs than throughput only.
3. PCKS: **systematically stress-test** your SFP system.
4. **Re-Use current benchmarks** to implement PCKS pattern.



Your questions and feedback are greatly appreciated!

http://openclipart.org/detail/16126/chasm-by-rygle

References

Anicic, D., Fodor, P., Rudolph, S., & Stojanovic, N. (2011). EP-SPARQL: a unified language for event processing and stream reasoning. In S. Srinivasan, K. Ramamritham, A. Kumar, M. P. Ravindra, E. Bertino, & R. Kumar (Eds.), Proceedings of the 20th international conference on World wide web - WWW '11 (pp. 635–644). New York, NY, USA: ACM.

Barbieri, D. F., Braga, D., Ceri, S., Della Valle, E., & Grossniklaus, M. (2010). C-SPARQL: A Continuous Query Language for RDF Data Streams. International Journal of Semantic Computing, 4(1), 3–25.

Jean-Paul Calbimonte, Óscar Corcho, Alasdair J. G. Gray: Enabling Ontology-Based Access to Streaming Data Sources. International Semantic Web Conference (1) 2010: 96-111

Cugola, G., & Margara, A. (2012). Processing flows of information. ACM Computing Surveys, 44(3), 1–62.

Daniele Dell'Aglio, Marco Balduini, Emanuele Della Valle, On the need to include functional testing in RDF stream engine benchmarks. Proc. Bersys 2013.

Gray, J. (1993). The Benchmark Handbook for Database and Transaction Systems (2nd ed.). Morgan Kaufmann.

Gray, Levine: "Thousands of DebitCredit Transaction per second: Easy and Inexpensive, MSR 2005-39…

Le-phuoc, D., Dao-tran, M., & Pham, M. (2012). Linked Stream Data Processing Engines : Facts and Figures. The Semantic Web - ISWC 2012: 11th International Semantic Web Conference, ISWC 2012, Boston, MA, USA, Nov 12-14, 2012, Proceedings (Vol. 1380, pp. 1–12).

Le-phuoc, D., Dao-tran, M., Parreira, J. X., & Hauswirth, M. (2011). A Native and Adaptive Approach for Unified Processing of Linked Streams and Linked Data. In L. Aroyo, C. Welty, H. Alani, J. Taylor, A. Bernstein, L. Kagal, N. Noy, et al. (Eds.), The Semantic Web -- ISWC 2011: 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I (Vol. 7031, pp. 370–388). Springer Berlin / Heidelberg.

Scharrenbach, T., Urbani, J., Margara, A., Valle, E. Della, & Bernstein, A. (2013). Seven Commandments for Benchmarking Semantic Flow Processing Systems. Proc.ESWC 2013 (pp. 1–15).

Tichy, W. F., Lukowicz, P., Prechelt, L., & Heinz, E. A. (1995). A Quantitative Evaluation Study in Computer Science : Journal of Systems and Software, 28(1), 9–18.

Wainer, J., Novoa Barsottini, C. G., Lacerda, D., & Magalhães de Marco, L. R. (2009). Empirical evaluation in Computer Science research published by ACM. Information and Software Technology, 51(6), 1081–1085. doi:10.1016/j.infsof.2009.01.002

Zhang, Y., Duc, P. M., Corcho, O., & Calbimonte, J. (2012). SRBench : A Streaming RDF / SPARQL Benchmark. The Semantic Web - ISWC 2012: 11th International Semantic Web Conference, ISWC 2012, Boston, MA, USA, Nov 12-14, 2012, Proceedings.

Zhang, Y & Boncz, Peter (2012). Benchmarking Linked Open Data Technology. Presentation at the European Data Forum (EDF) 2012, Copenhagen.