Optique

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF OXFORD

Towards Analytics Aware Ontology Based Access (OBDA) to Static and Streaming Data

Evgeny Kharlamov
Senior Research Fellow
Department of Computer Science
University of Oxford

Y. Kotidis, T. Mailis, C. Neuenstadt, C. Nikolaou
Ö. Özcep, C. Svingos, D. Zheleznyakov, S. Lamparter
I. Horrocks, Y. Ioannidis, R. Möller

ΟΠΑ AUEB

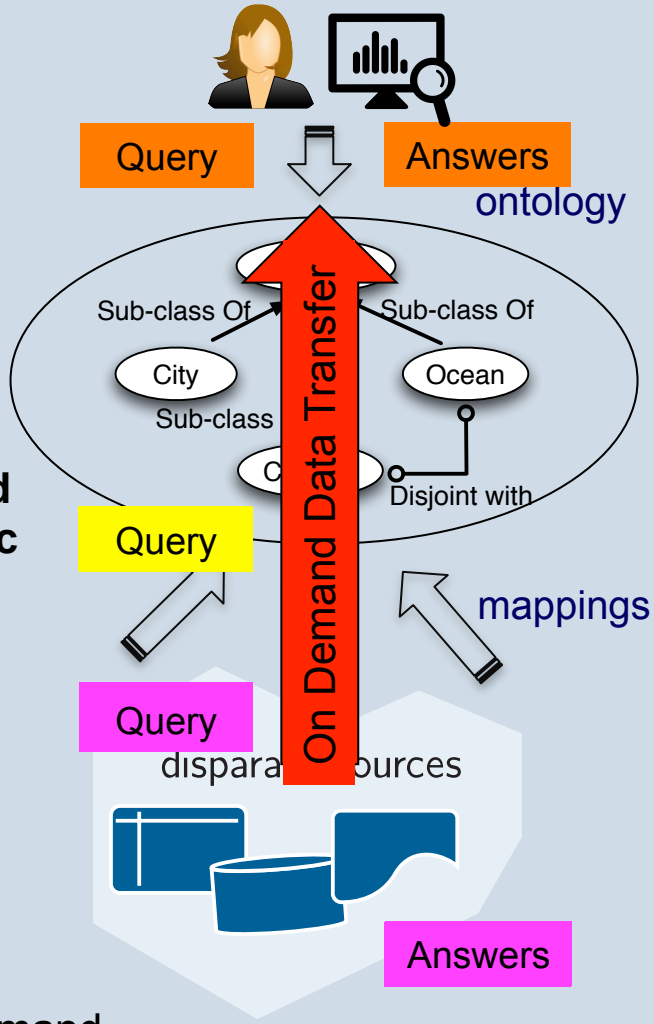
HELLENIC REPUBLIC
National and Kapodistrian University of Athens

SIEMENS

Ontology Based Data Access

Main Idea

- an approach to Data integration
- ontology
  - provides a common schema over several DBs
  - mediates data and data consumers
- mappings
  - "connect" ontology and DBs
- virtual data integration approach
  - data stays where it was
  - data transferred to consumers on demand
  - automatic query processing: semantic queries → data queries



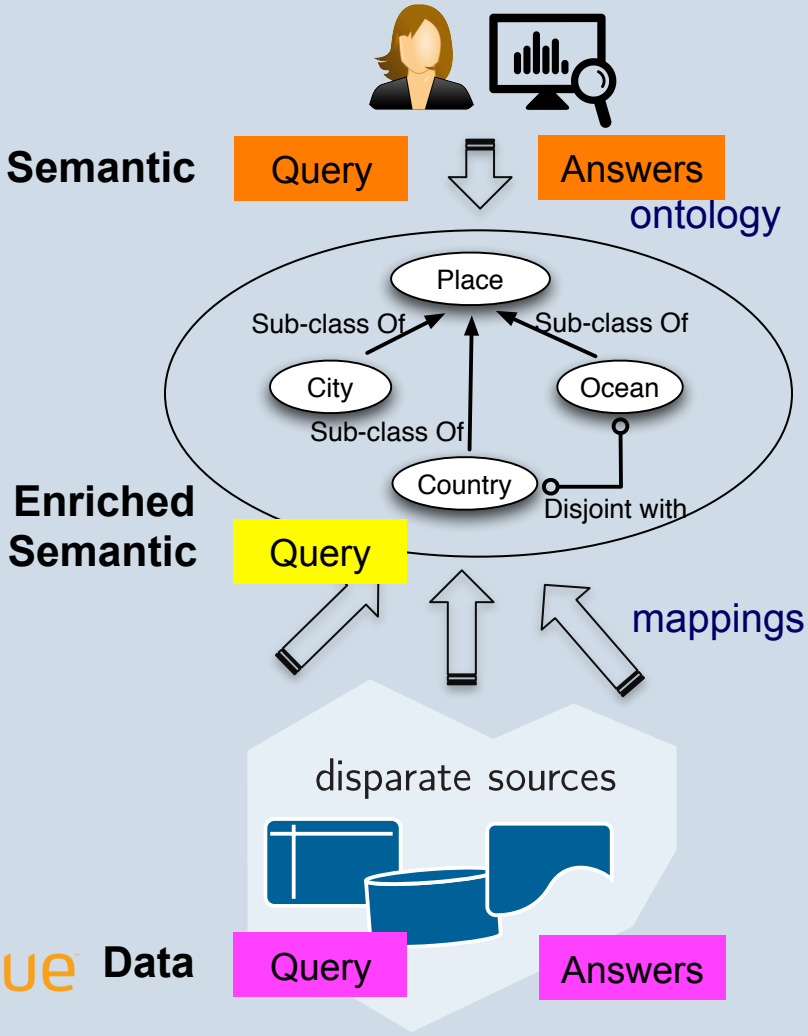
Success Stories

OBDA systems

- Include:
  D2RQ, Mastro, morph-RDB, Ontop,
  OntoQF, Ultrawrap, Virtuoso, Optique
- Implement from some
  to all OBDA features

OBDA has been applied

- cultural heritage
- governmental organizations
- Industry
  - Statoil
  - Siemens
  - E-commerce



Statoil SIEMENS } Optique

Rakuten

Where are the OBDA limits?

Hard case for OBDA

Siemens turbine monitoring example:

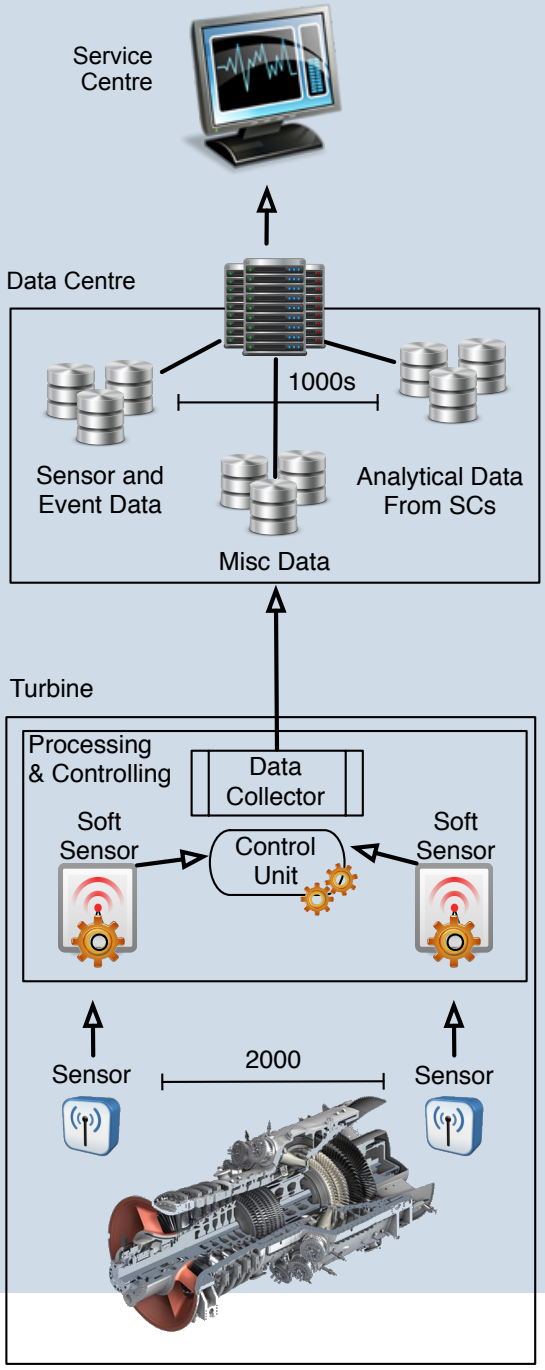
- Detect alerting temperature behavior
- Can be done by queries that ask for
  - reliable sensors reporting alerting temp. (patterns)

Terminology

- Reliable: good avg. score (>=90%) of validation tests
- Alerting: similar to what we saw last year when there was an alerting situation
- Similar: Pearson correlated by at least 0.75

In a given turbine report all temperature sensors that are reliable, i.e., with the average score of validation tests at least 90%, and whose measurements within the last 10 min were similar, i.e., Pearson correlated by at least 0.75, to measurements reported last year by a reference sensor that had been functioning in a critical mode.

Q: return reliable sensors reporting alerting temp.



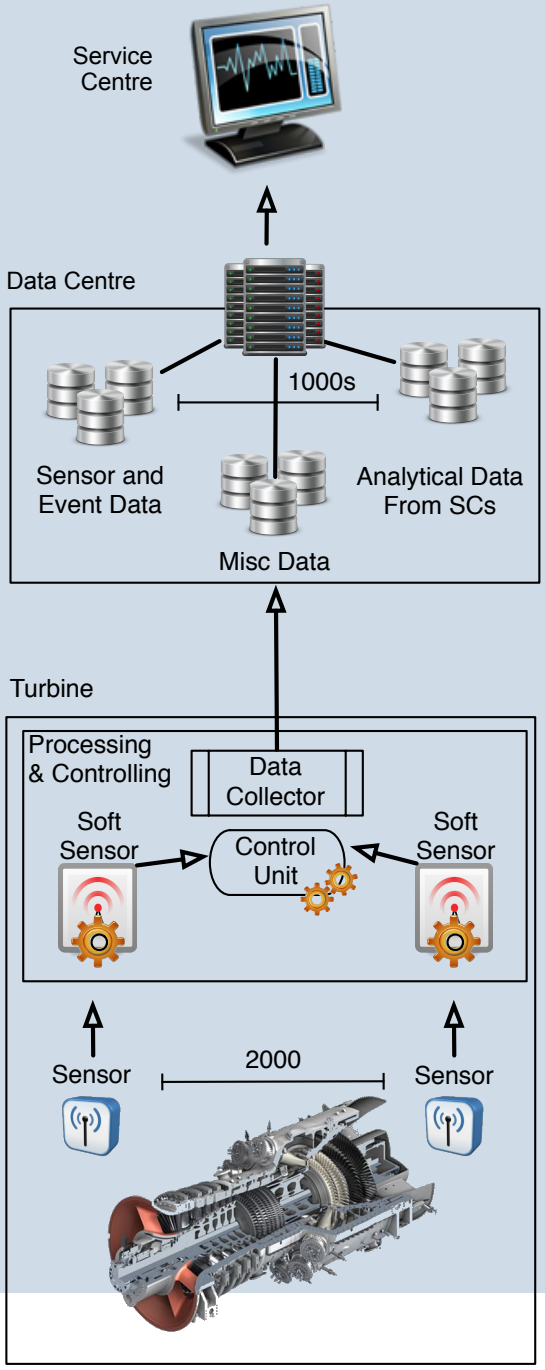
Hard case for OBDA

Important components of the query

- access to both static and streaming data
  - Streaming: data currently produced by sensors
  - Static: historic from last year; turbine structural data
- analytics
  - Average score
    - aggregate function
  - At least 90%
    - Value comparison
  - Pearson correlation

In a given turbine report all temperature sensors that are reliable, i.e., with the average score of validation tests at least 90%, and whose measurements within the last 10 min were similar, i.e., Pearson correlated by at least 0.75, to measurements reported last year by a reference sensor that had been functioning in a critical mode.

Q: return reliable sensors reporting alerting temp.



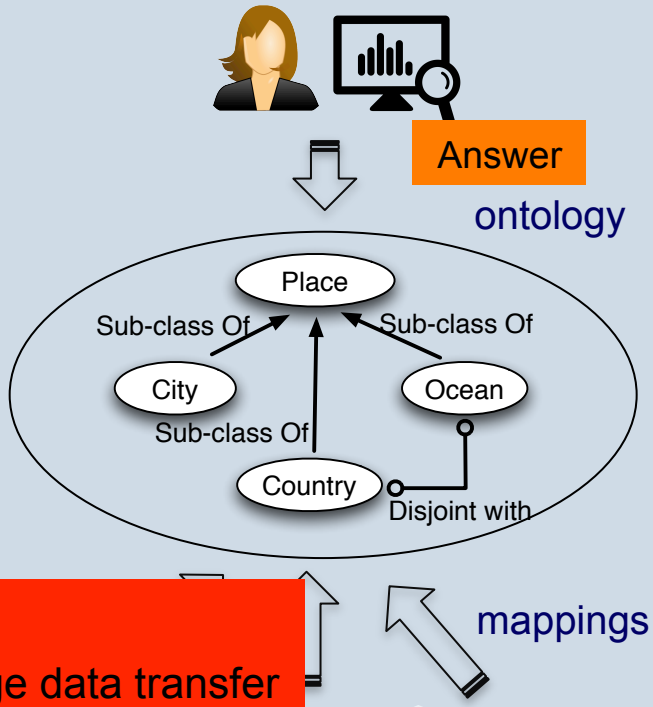
Existing OBDA

Support

- static relational data/queries
- streaming data/queries
- but not both

Are good for conjunctive queries but not for

- analytics
  - epistemic semantic: analytics over “semantic” answers
  - encoding in mappings



Inefficient

- Requires large data transfer
- Ignore source capabilities

brittle and inflexible

- Requires encoding of specific values in mappings
- query dependent

In a given turbine report all temperature sensors tha
the average score of validation tests at least 90%, an
within the last 10 min were similar, i.e., Pearson correl
measurements reported last year by a reference sensor t
in a critical mode.

Q: return reliable sensors reporting alerting temp.

OBDA 2.0: for a Big company

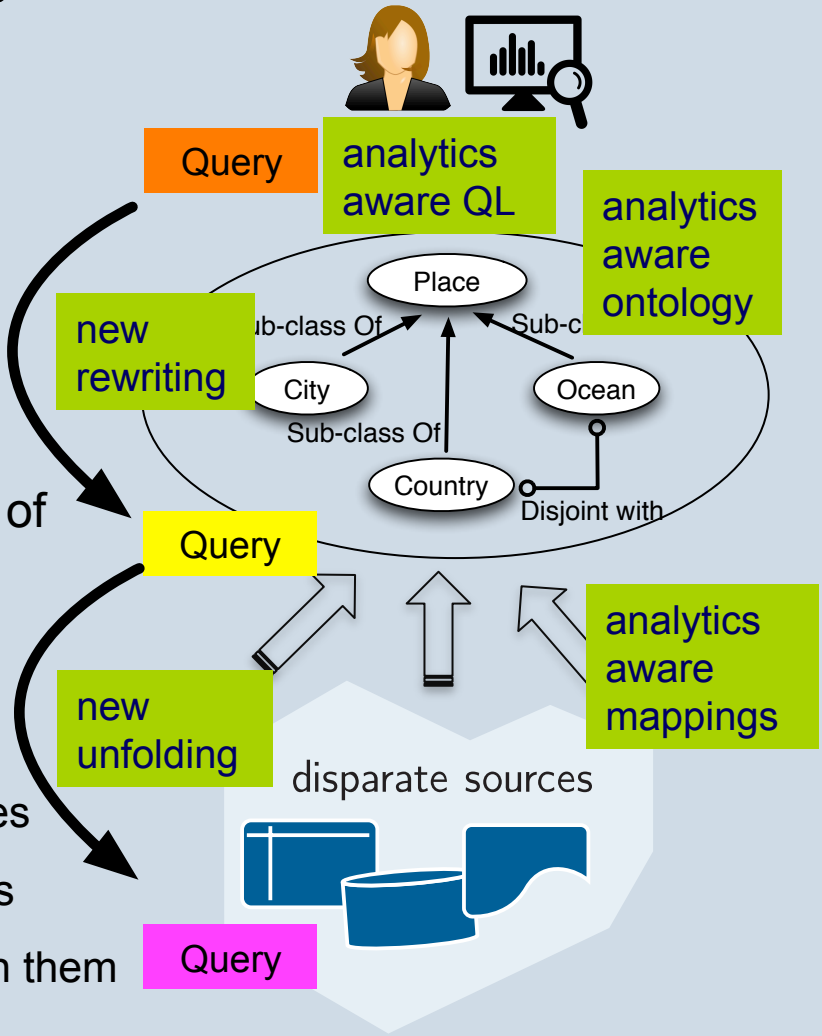
Big company

- data processing is analytics oriented
- has DBs of various kinds

Should become: Analytics Aware

- by supporting declarative representations of basic analytics operations
- Requires:
  - analytics aware
    - – ontologies, mappings, query languages

      capable of capturing analytical operators
  - new Q. processing techniques to cope with them
    - – rewriting, unfolding

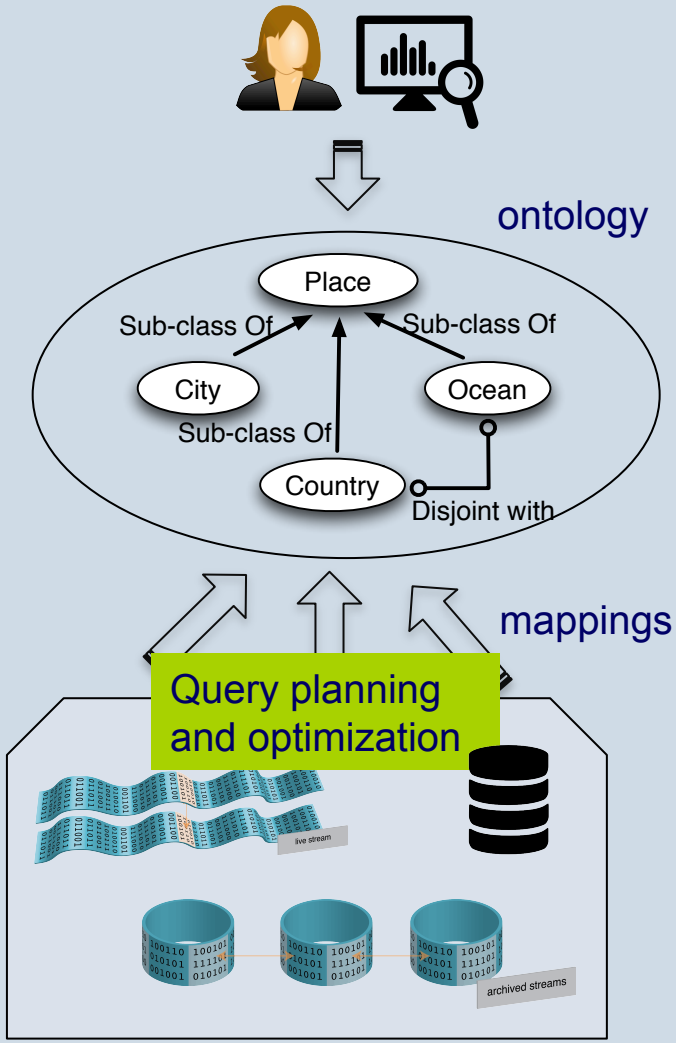


OBDA 2.0: for a Big company

Should become: Source and Cost Aware

- by supporting data sources of various types
  - live and archived streams
  - relational DBs
  - other DBs
- by offering a robust
  - query planning
  - query optimization

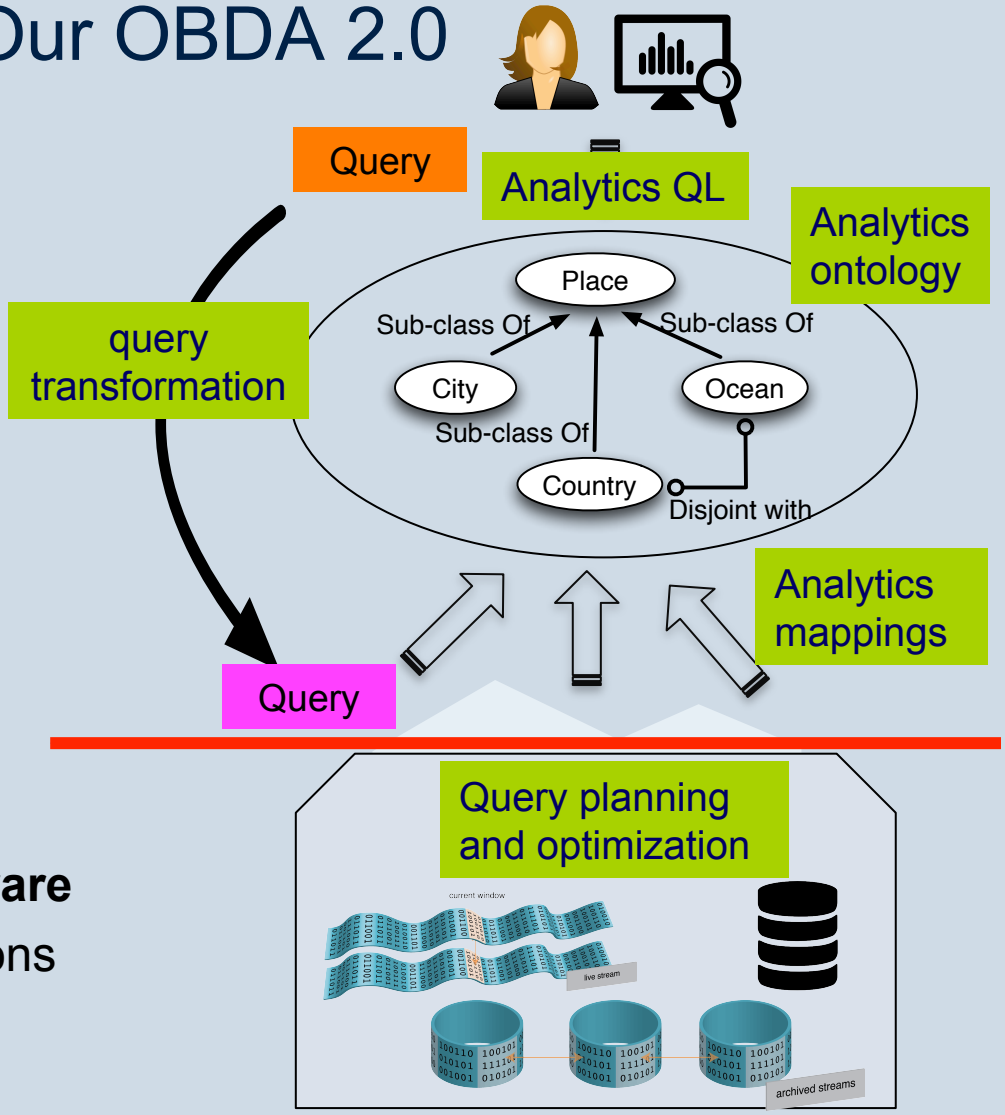
  for estimating the cost of different plans,
  and use such estimates to produce low-cost plans



Outline: Overview of Our OBDA 2.0

Analytics Aware

- Ontologies
- Queries
- Mappings
- Query transformation
  - Ontology → Data queries



Analytics, Source, and Cost Aware

- Query planning and optimisations
- Experiments

Analytics Aware Ontology: DL-Lite^{agg}

Main Ideas

- Why DL-Lite: classical DL-Lite is designed for OBDA
- Extensions
  - Concepts that are based on aggregation of Att values

$$\geq_{0.9} (\min\ testScore) \sqsubseteq Reliable,$$

- Semantics
  - Closed-world semantics for new (aggregate) concepts
  - Open-world semantics is meaningless
- Conjunctive query answering
  - Tractable when aggregate function computation is tractable
  - Thanks to closed-world interpretation of predicates involved in aggregation [1]

$$B \rightarrow A \mid \exists R, \quad C \rightarrow B \mid \exists F, \quad E \rightarrow \circ_r(\text{agg}\ F), \quad R \rightarrow P \mid P^{-},$$

[1] C. Lutz, I. Seylan, and F. Wolter. Mixing Open and Closed World Assumption in Ontology- Based Data Access: Non-Uniform Data Complexity. DL. 2012.

Analytics and Source Aware QL: STARQL

```
1  PREFIX ex : <http://www.siemens.com/onto/gasturbine/>
2
3  CREATE PULSE examplePulse WITH START = NOW, FREQUENCY = 1min
4
5  CREATE STREAM StreamOfSensorsInCriticalMode AS
6  CONSTRUCT GRAPH NOW { ?sensor a :InCriticalMode }
7
8  FROM STATIC ONTOLOGY ex:sensorOntology, DATA ex:sensorStaticData
9  WHERE { ?sensor a ex:Reliable }
10
11 FROM STREAM   sensorMeasurements          [NOW - 1min, NOW]-> 1sec
12              referenceSensorMeasurements 1year <-[NOW - 1min, NOW]-> 1sec,
13 USING PULSE   examplePulse
14 SEQUENCE BY   StandardSequencing AS MergedSequenceOfMeasurementes
15 HAVING EXISTS i IN MergedSequenceOfMeasurementes
16       (GRAPH i { ?sensor ex:hasValue ?y. ex:refSensor ex:hasValue ?z })
17       HAVING PearsonCorrelation(?y, ?z) > 0.75
```

- Input
  - Static analytics aware ontology and data set
  - collection of streams: life and archived
- Output
  - Stream of data sets

Conjunctive queries

Diagnostic queries
- standard agg: count, avg
- advanced agg: Pearson correlation

Query Transformation

Process Overview:

$$Q_{\text{starql}} \approx Q_{\text{StatCQ}} \wedge Q_{\text{Stream}} \xrightarrow[\mathcal{O}]{\text{rewrite}} Q'_{\text{StatUCQ}} \wedge Q'_{\text{Stream}}$$

$$\xrightarrow[\mathcal{M}]{\text{unfold}} Q''_{\text{AggSQL}} \wedge Q''_{\text{Stream}} \approx Q_{\text{sql}^{\oplus}}$$

- Rewriting
  - Essentially: standard perfect reformulation algorithm for DL-Lite
- Unfolding
  - Relies on mappings of 2 kinds
    - Standard concepts
    - Aggregate concepts

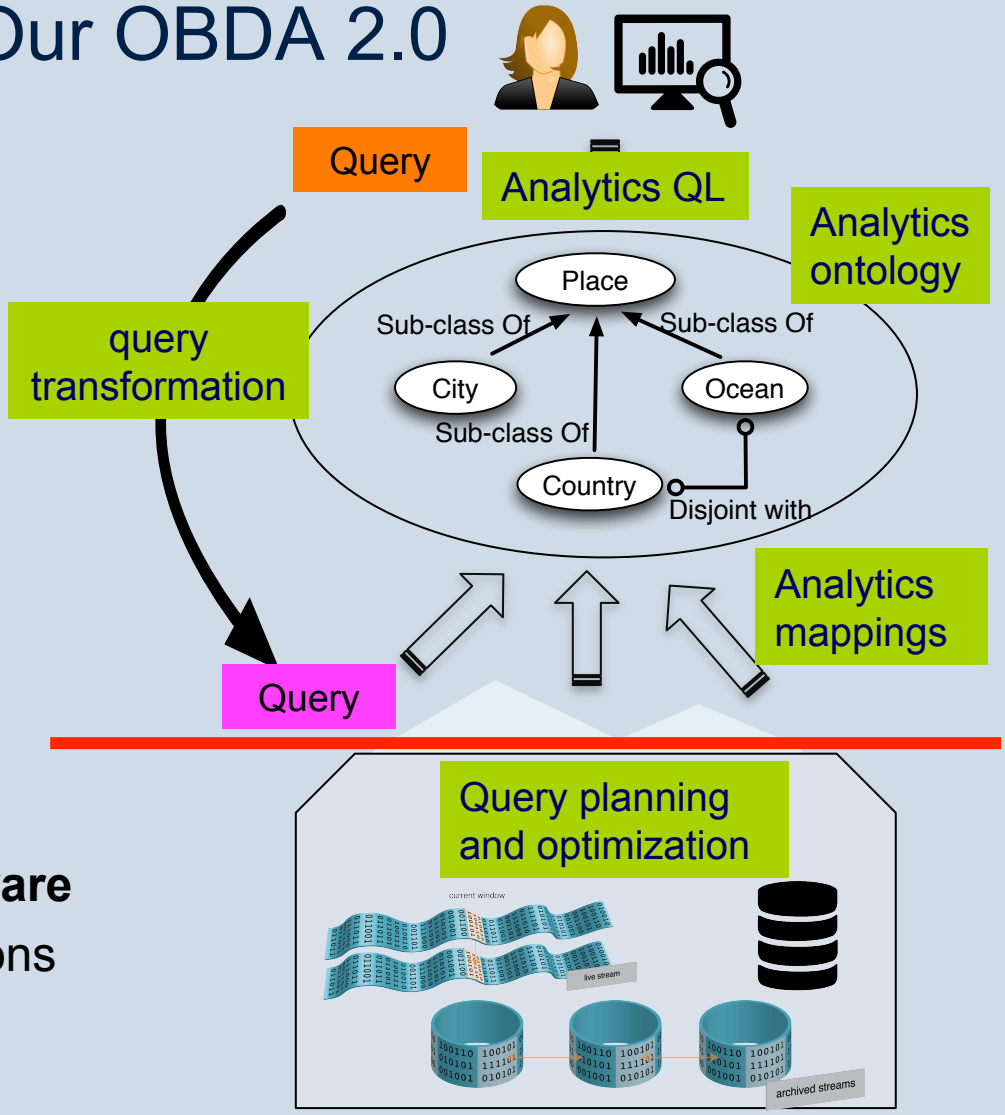
Mapping for aggregate concepts:

$$\big(\geq_{0.9} (\text{min } testscore)\big)(x) \leftarrow \begin{array}{ll} \text{SELECT} & x \\ \text{FROM} & \text{unfold}(\text{rewrite}(testscore(x, y))) \\ \text{GROUPBY} & x \text{ HAVING min}(y) \geq 0.9 \end{array}$$

Outline: Overview of Our OBDA 2.0

Analytics Aware

- Ontologies
- Queries
- Mappings
- Query transformation
  - Ontology → Data queries



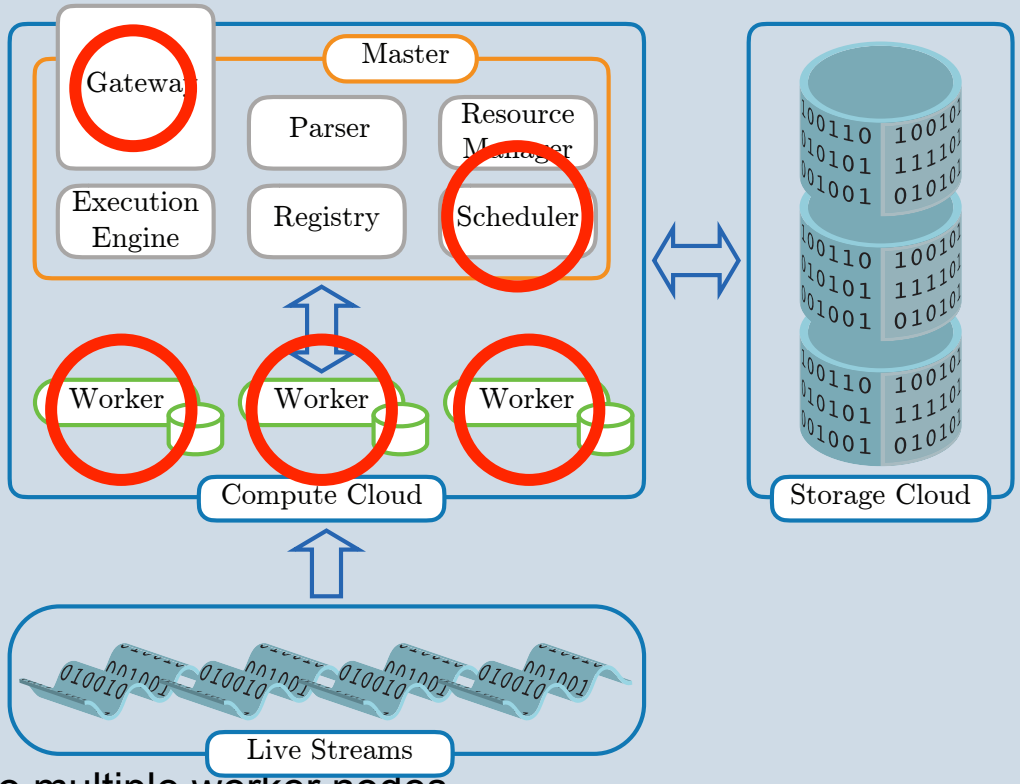
Analytics, Source, and Cost Aware

- Query planning and optimisations
- Experiments

Optimization of Data Queries: ExaStream

Exastream: Data-Stream Management System

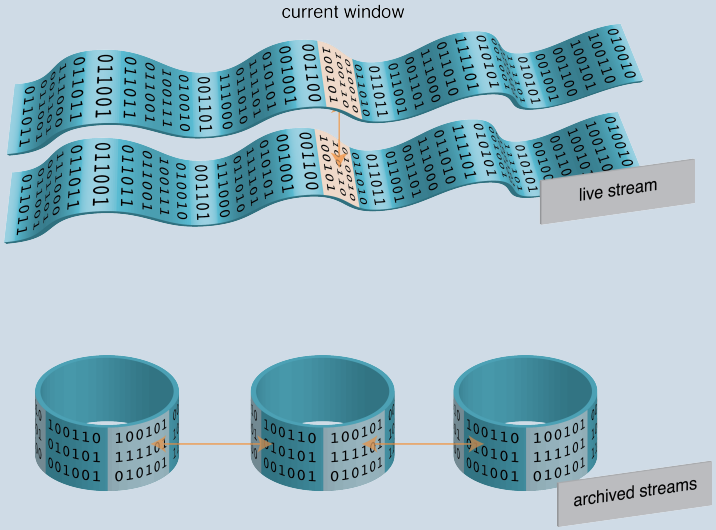
- our streaming extension
  - of SQLite DB engine
- SQL⊕ queries
  - SQL enhanced with
  - User Def. Func. (for analytics)
- smart query planner based on
  - query
  - available stream/static DBs
  - execution environment
- parallelism and distribution on a cloud
  - to accelerate analytics
  - distribution of queries & data to multiple worker nodes
- Query execution cycle
  - Q registers at Gateway Server → parsed and fed to the Scheduler
  - Scheduler places Qs on available Workers and optimizes their execution



Our Query Optimisations

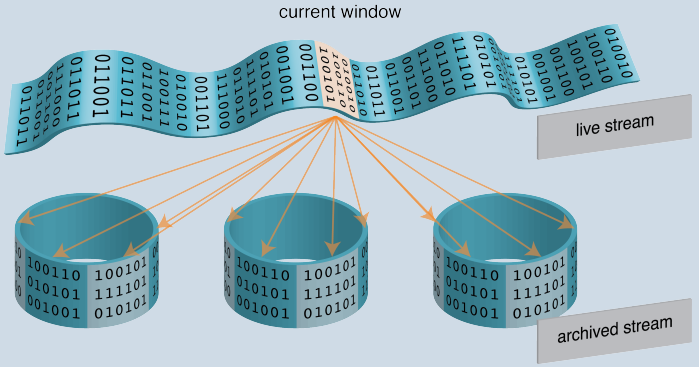
Analytical operations of 3 types

- live-stream operations
  - analytical tasks on live streams
- static-data operations
  - analytical tasks on static information
- hybrid operations
  - analytical tasks on live-streams & static data

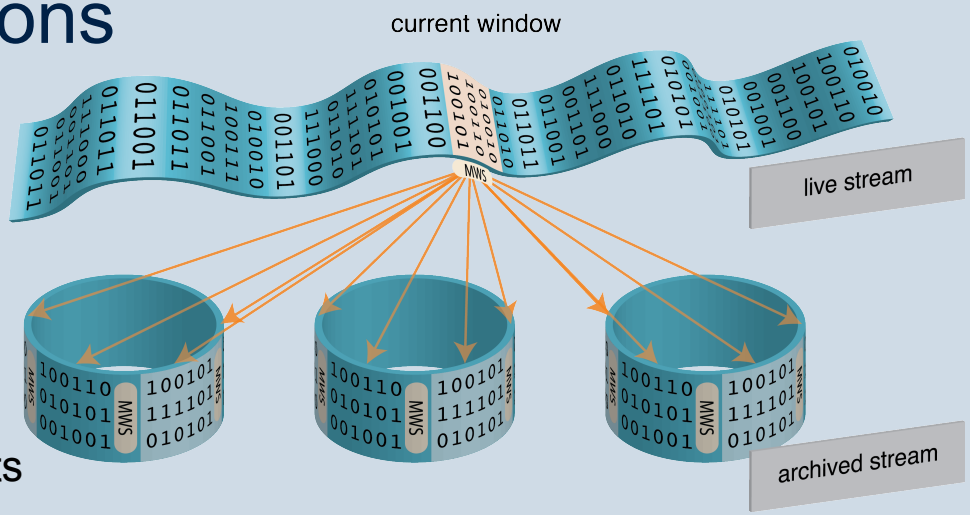


Basic Optimizations & strategies

- data views
  - compute static analytical tasks ahead of live-stream operations
- adaptive indexing
  - adaptively building main-memory indexes on batches of cached stream tuples



View Based Optimisations



Ex: pre-computation of AVG for streaming analytics

- Measurements
  - raw data
  - archived part of a data stream
  - has time & actual measurements
- Windows
  - pre-computed data
  - stores the windowing mechanism
  - has window-id, starting, and ending point
  - has frequently asked aggregates, e.g., AVG

Windows

Wid	MWS_Avg	Window_Start	Window_End
1	427ºC	2016-02-08, 15:00:00	2016-02-08, 15:01:00
2	440.5ºC	2016-02-08, 15:02:00	2016-02-08, 15:03:00

Measurements

Time	Measurment
2016-02-08, 15:00:00	426ºC
2016-02-08, 15:01:00	428ºC
2016-02-08, 15:02:00	433ºC
2016-02-08, 15:03:00	448ºC

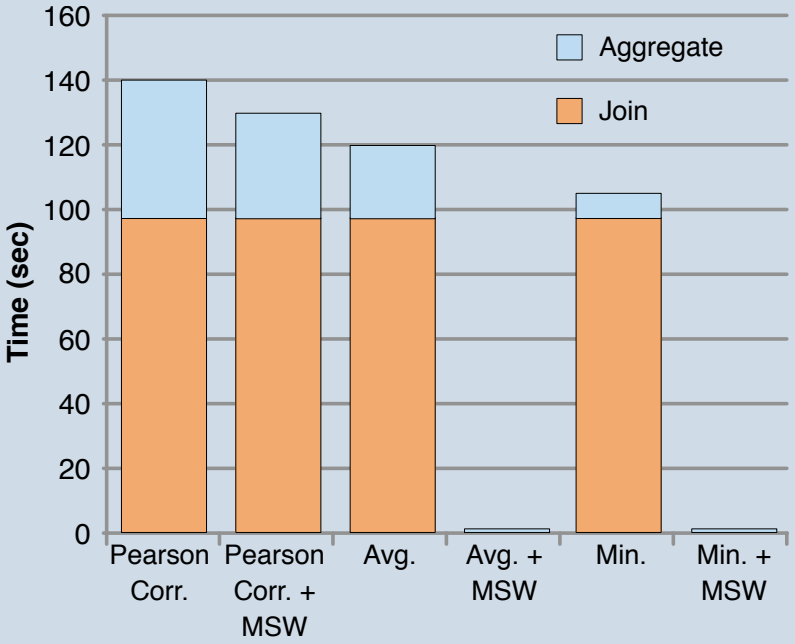
Evaluation

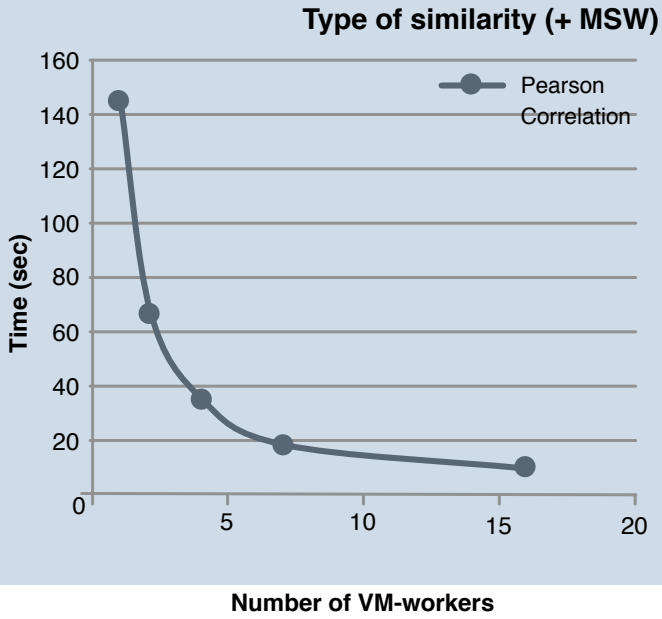
View Based Optimization (1 node)

- measured
  - time to process a live-stream
  - against 100,000 archived ones
  - time is divided
    - to join-time & aggregate time
- Outcome
  - for Pearson correlation
    - time gain: 8.18%
  - for AVG and MIN
    - optimizer prunes many unnec. joins

Intra-query Parallelism (1-16 nodes)

- distribution of pre-computed views among nodes
- significant time decrease





Summary

Discussed

- hard cases for OBDA
  "Siemens real-time turbine diagnostics"
- for them we need OBDA 2.0
  - analytics awareness
  - source and cost awareness

Introduced

- Example OBDA 2.0
- Experiments