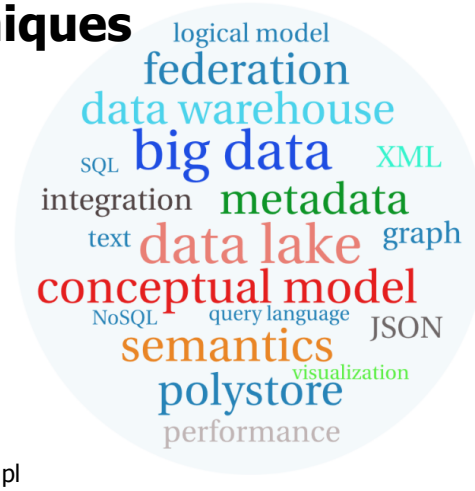
Novel Big Data Integration Techniques - What is New



Robert Wrembel
Robert.Wrembel@cs.put.poznan.pl
www.cs.put.poznan.pl/rwrembel
POZNAN UNIVERSITY OF TECHNOLOGY

Landscape: past

- **Data models**
  - **relational**
  - **object-oriented**
  - **semi-structured**
  - **...**
- **Data formats**
  - **numbers, dates, strings**
  - **...**
- **Veolocity**
  - **OLTP systems**

n*V

- **Volume**
- **Velocity**
- **Variety**
  - data formats
  - 80% - 90% of the world's data is now unstructured
- **Veracity** (quality, reliability)
- **Value**
- **Variability** (changes in values or meaning)
- **Visualization**

- **frequently changing (e.g., Facebook)**
- **constantly changing (streams)**

- **Data models**
  - **relational**
  - **graphs**
  - **NoSQL**
  - **semi-structured**
  - **...**
- **Data formats**
  - **numbers, dates, strings**
  - **HTML, XML, JSON**
  - **time series and sequences**
  - **texts**
  - **multimedia**
  - **...**

Needs

- **Performance**
  - **storing efficiently (fast writes, compression)**
  - **retrieving efficiently (fast scans, fast selective search)**
- **Integration**
- **Understanding structure and content**
- **Querying and presenting**

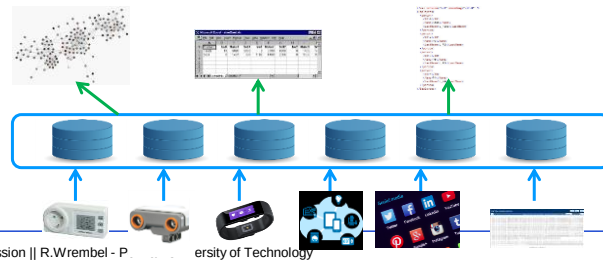
Data integration

- Past
  - physical: data warehouse
  - virtual: federated/mediated

- Today
  - physical: data lake
  - virtual: federated/mediated
  - mixed: polystore

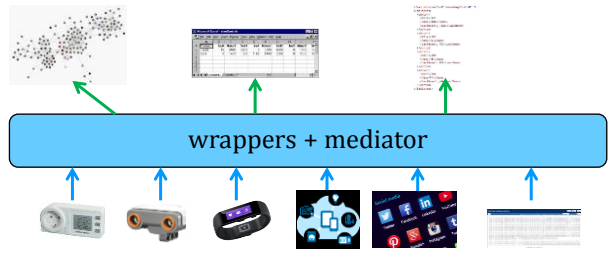
Big Data integration

- **Physical** integration → data lake
  - large repository of heterogeneous data (in multiple native data models/formats)
  - no schema on write - schema on read
  - typically based on a distributed file system
  - need for refreshing
    - how to detect changes?
    - even incremental refreshing uploads large volumes of data (new algorithms for incremental refreshing?)



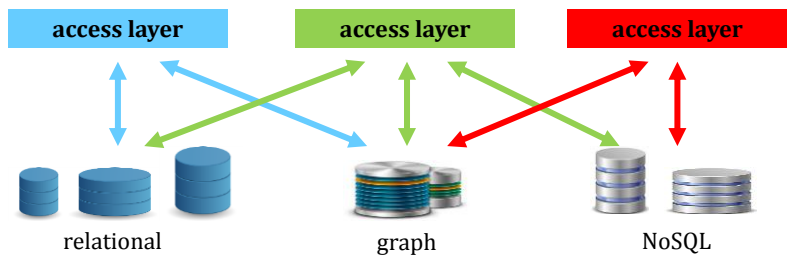
Big Data integration

- **Virtual → analogy to mediated/federated architectures**



Big Data integration

- **Mixed architecture → polystore**
  - **federation of islands of information**
  - **island of information: collection of storage engines using the same data model (query language)**



J. Duggan, A.J. Elmore, M. Stonebreaker, et. al.: The BigDAWG Polystore System. SIGMOD Record, Vol. 44, No. 2, 2015

Big Data integration challenges

- Need to understand data sources
  - structures
  - content
- Rich and understandable metadata describing data sources
  - automatic or semi-automatic metadata ingestion from new data sources plugged into an integration system
  - efficient metadata storing, searching, and visualizing

Big Data integration challenges

- Which integrated model (conceptual and logical) to use?
  - data sources: graphs + semi-structured + relational + NoSQL + ...
  - how to construct it
- Schema design methods
  - automatic discovery?

Big Data integration challenges

- How to (semi)-automatically discover data sources?
  - DS structure discovery
  - DS content understanding
- How to dynamically plug-in a DS into a federation?

Big Data integration challenges

- Efficient ETL/ELT architectures for ingesting data into a DL and further, for producing clean and more structured data
  - a Big Data ETL engine processes much more complex ETL/ELT workflows and much larger data volumes than a standard one, the performance optimization of the engine becomes vital
  - ETL workflows often require UDFs, whose optimization is difficult

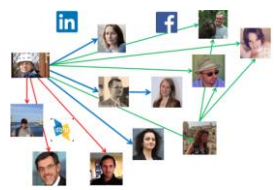
Big Data integration challenges

- Query processing (virtual, DL)
  - finding relevant data sources for a query
    - relevant "schema"/structure
    - relevant content
    - correlating multiple data sources of the same semantics
    - selecting the most reliable data sources
  - finding the relevant data sources quickly
  - parsing, decomposing, translating into native dialects, and routing
  - efficiently integrating (transform, clean, de-duplicate, integrate) on the fly data returned by local queries

Big Data integration challenges

- Developing a query language capable of handling data complexity, heterogeneity, and incompleteness
  - user preferences in a query: QoS, QoD, an output data format, visualization prefs

find colleagues of Robert Wrembel
context business, research, social
output graph | table



Big Data integration challenges

- Virtual data integration architectures
  - advantage
    - access to up-to-date data
    - more suitable for dynamically changing DSs
  - pittfal
    - slow → query resolving and data integration is executed on the fly
- New optimization techniques are needed
  - caching the results at two levels: in main memory and on disk
  - deciding what to cache
  - deciding which queries should be executed on data sources and which on cached data
  - proactive cache refreshing

Big Data integration challenges

- Extensive usage of metadata
  - schema/structure
  - content
  - transformation rules
  - visualization
  - performance
  - ...
- Metadata modeling/standard
  - CWM for relational systems
  - ? for Big Data architectures

Big Data integration challenges

- **Comprehensive mechanisms for Big Data provenance**
  - **gathering**
  - **representation**
  - **maintenance**
  - **storage**
  - **querying**
  - **augmenting user queries with the most relevant provenance data**
    - **info about which DSs were used to answer a query**
    - **info about the quality of the used DSs**

Some references

- S.M.F Ali, R. Wrembel: From conceptual design to performance optimization of ETL workflows: current state of research and open problems. The VLDB Journal, 2017, DOI 10.1007/s00778-017-0477-2
- P. Ceravolo, et. al.: Big Data and Data Semantics: Three Levels of Enhancement. To appear in Journal on Data Semantics, Springer, 2018
- J. Stefanowski, K. Krawiec, R. Wrembel: Exploring Complex and Big Data. To appear in Int. Journal of Applied Mathematics and Computer Science, de Gruyter, 2018