POZNAN UNIVERSITY OF TECHNOLOGY

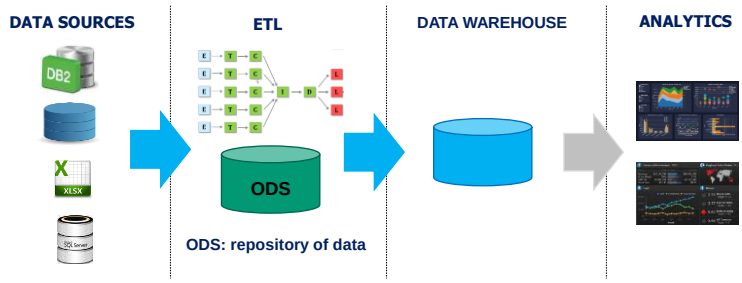
On Warehousing Standard and Big Data

Robert Wrembel
Poznan University of Technology
Institute of Computing Science
Poznań, Poland
Robert.Wrembel@cs.put.poznan.pl
www.cs.put.poznan.pl/rwrembel

Outline

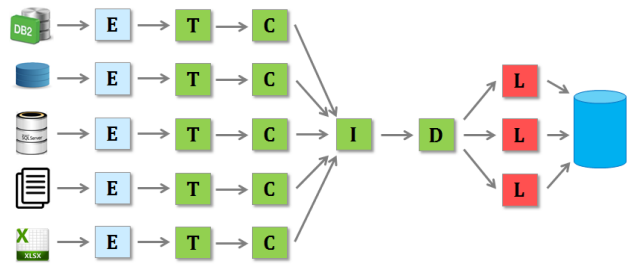
- Data warehouse architecture and ETL
- Data source evolution
  - ETL evolution
  - data warehouse evolution
- ETL optimization
- Data processing architecture
- Data integration project @PKO BP

Traditional DW architecture



- Goal: to integrate data in a unified format
- DSs are integrated by the ETL layer

ETL



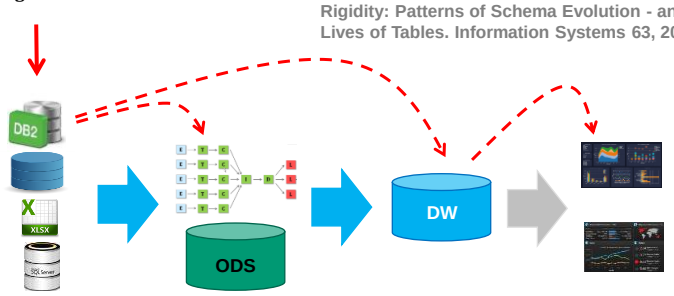
- Extraction - Transformation - Loading
  - Data Processing Pipeline
  - Data Processing Workflow
- ETL processes are composed of sequences of tasks

- ETL processes
  - ingesting
  - transforming
  - cleaning
  - homogenizing
  - de-duplicating
  - uploading

Evolving data sources

add column
drop column
change column datatype
change column size
create table
drop table
rename column
rename table
split table
merge tables

- D. Sjøberg : Quantifying schema evolution. IST 35(1), 1993
- C.A. Curino, L. Tanca, H.J. Moon, C. Zaniolo: Schema evolution in wikipedia: toward a web information system benchmark. ICEIS, 2008
- H. J. Moon, C. A. Curino, A. Deutsch, C.-Y. Hou, C. Zaniolo: Managing and querying transaction-time databases under schema evolution. VLDB, 2008
- P. Vassiliadis, A. V. Zarras. 2017. Schema Evolution Survival Guide for Tables: Avoid Rigid Childhood and You're en Route to a Quiet Life. Journal on Data Semantics 6(4), 2017
- P. Vassiliadis, A. V. Zarras, I. Skoulis. 2017. Gravitating to Rigidity: Patterns of Schema Evolution - and its Absence - in the Lives of Tables. Information Systems 63, 2017



Impact on ETL

- **Deployed ETL process (workflow) may no longer be executed → needs to be repaired**
- **Pharma and banks**
  - **# data sources integrated → from dozens to over 200**
  - **# deployed workflows → from thousands to hundreds of thousands**
- **Goal → (semi-)automatic ETL process repair**

Problem solving?

- Neither commercial nor open software support (semi-)automatic repair
  - only impact analysis is supported
- Business architectures
  - writing generic ETL
    - input to a generic ETL: tables and attributes
  - screening DS changes
    - kind of views
- Manual repairs are needed

Research approaches

- Evolution (repair) rules defined for some tasks of an ETL process
  - G. Papastefanatos, P. Vassiliadis, A. Simitsis, T. Sellis, Y. Vassiliou: Rule-based Management of Schema Changes at ETL sources. ADBIS, 2010
  - P. Manousis, P. Vassiliadis, G. Papastefanatos: Impact Analysis and Policy-Conforming Rewriting of Evolving Data-Intensive Ecosystems. Journal on Data Semantics, 4(4), 2015
  - D. Butkevicius, P.D. Freiberger, F.M. Halberg, J.B. Hansen, S. Jensen, M. Tarp: MAIME: A Maintenance Manager for ETL Processes. EDBT/ICDT Workshops, 2017
- Drawbacks
  - rules must be explicitly defined for each graph element: huge overhead for complex processes
  - a user must determine a policy in advance, before an evolution event occurs
  - limited to tasks expressed by SQL

Research approaches

- **Case based reasoning**
  - A. Wojciechowski: ETL workflow reparation by means of case-based reasoning. Information Systems Frontiers 20(1), 2018
- **Rules discovery from CBR**
  - J. Awiti: Algorithms and Architecture for Managing Evolving ETL Workflows. Proc. of ADBIS Workshops, Springer CCIS 1064, 2019
  - J. Awiti, R. Wrembel: Rule Discovery for (Semi-)automatic Repairs of ETL Processes. DB&IS, Springer CCIS 1243, 2020
- **Drawbacks**
  - **a library of cases is needed**
  - **the correctness of a proposed repair cannot be formally checked**
- **All these approaches solve only a fraction of the problem**
- **Big data sources make the problem worse**

Case study

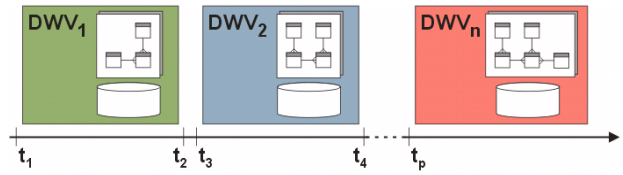
- **A pilot project for bank PKO BP**
- **Collecting data from web portals**
  - **offers of other banks**
  - **data on companies (LinkedIn, Glassdor, open data of public administration)**
- **Data ingested into a repository**
  - **Amazon Web Services**
  - **Google Cloud Platform**

Case study

- Main problems
  - frequent (often day-to-day) changes in structures of web pages
    - parsing is challenging
  - frequent changes in values
    - e.g., saving plan → interest rate
    - a need to analyze temporal data
  - old problem of evolving DSs
  - structural changes much frequent than in a standard DW architecture

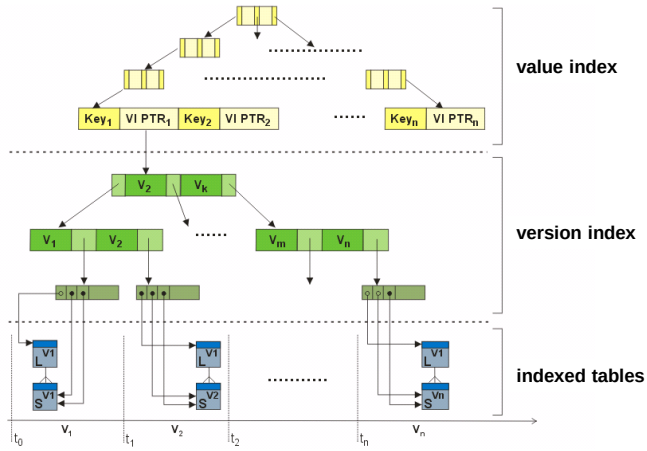
Multiversion DW

- Handling
  - data changes, esp. dimension instance
  - schema changes
- MVDW: a sequence of DW versions
  - a schema version (facts, dimensions)
  - an instance version (stores the set of data consistent with its schema version; measures/cell values; dimension data)



Multiversion DW

- **Querying: MVQL**
- **Indexing: Multiversion Join Index**



Versioning in Data Lakes

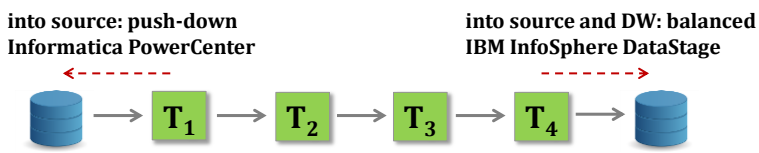
- **Version management → requirement stated in:**
  - F. Nargesian, E. Zhu, R.J. Miller, K.Q. Pu, P.C. Arocena: Data Lake Management: Challenges and Opportunities. VLDB Endow., 2019

ETL performance optimization

- **Goal**
  - to build an ETL workflow execution plan
  - to optimize the plan
  - like SQL query optimization
- **Plan optimization**
  - discovering and removing redundant parts
  - building a cost model
    - data statistics → processed data sets are not available in advance, time overhead for computing them
    - performance statistics
  - tasks reordering

Commercial approaches

- **Increasing resources (#CPU, memory, #nodes)**
- **Moving tasks to decrease data volume asap**
  - constrained to tasks expressed by SQL



- **How to automatically implement an efficient ETL task in a DS (relational, non-relational)?**
  - query optimizer, indexes, partitioning, CS/RS, ...
- **Project with IBM Software Lab Kraków**

- On Evaluating Performance of Balanced Optimization of ETL Processes for Streaming Data Sources. DOLAP, 2020

Commercial approaches

- **Parallelizing ETL tasks → running ETL in a cluster**
  - **IBM InfoSphere DataStage**
  - **Informatica PowerCenter**
  - **AbInitio**
  - **Microsoft SQL**
  - **Server Integration Services**
  - **Oracle Data Integrator**

Commercial approaches

- **Which tasks to parallelize?**
- **What is an optimal number of parallel processes?**
- **What is an optimal amount of resources (nodes, CPU, memory, threads)?**
- **Which parallelization skeleton to apply?**

ETL optimization: research

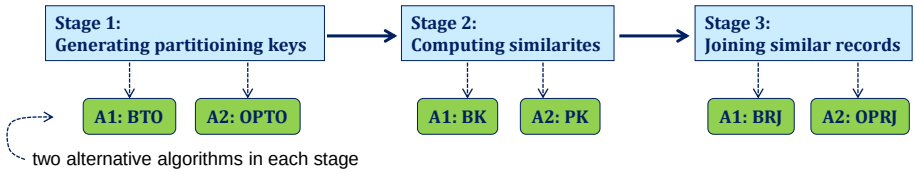
- **Methods review:**
  - S. M. F. Ali, R. Wrembel: From conceptual design to performance optimization of ETL workflows: current state of research and open problems. VLDB Journal 26(6), 2017
- **Workload partitioning and parallelization**
  - X. Liu, N. Iftikhar: An ETL Optimization Framework Using Partitioning and Parallelization. SAC, 2015
  - **partitioning methods into so-called execution trees**
    - **vertical**
    - **horizontal**
    - **single task partitioning and multi-threading**

ETL optimization: research

- **Performance optimization by task reordering → exponential complexity**
  - A. Simitsis, P. Vasiliadis, T. Sellis: State-Space Optimization of ETL Workflows. IEEE TKDE 17(10), 2005
    - **heuristics for reordering**
  - N. Kumar, P.S. Kumar: An Efficient Heuristic for Logical Optimization of ETL Workflows. BIRTE, 2010
    - **focuses on optimizing linear flows only**

Case study

- **Set-similarity joins using MapReduce (SSJ-MR)**
  - R. Vernica, M. Carey, C. Li: Efficient parallel set-similarity joins using MapReduce. SIGMOD, 2010



- **Each stage of SSJ-MR may be executed in a differently configured Amazon EMR cluster**

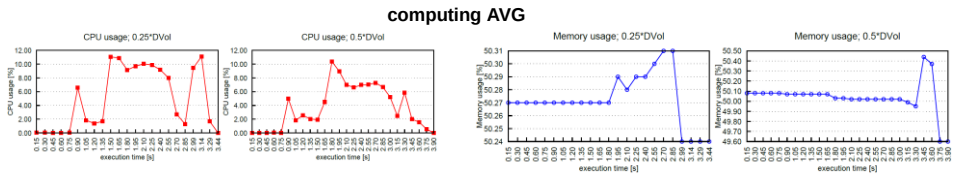
		#Nodes [exec cost/h]			
Stage	Algorithm	2 nodes @[0.4$/h]	4 nodes @[0.8$/h]	8 nodes @[1.6$/h]	10 nodes @[2.0$/h]
1	BTO	191.98	125.51	91.85	84.02
	OPTO	175.39	115.36	94.82	92.80
2	BK	753.39	371.08	198.70	164.57
	PK	682.51	330.47	178.88	145.01
3	BRJ	255.35	162.53	107.28	101.54
	OPRJ	97.11	74.32	58.35	58.11

Case study

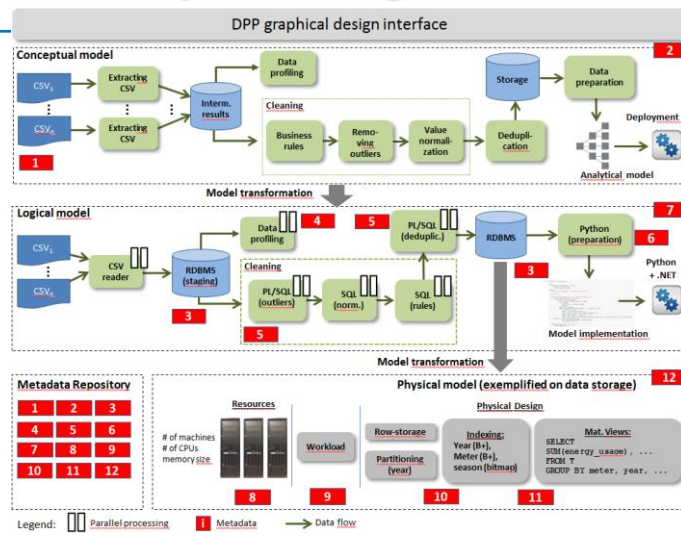
- **General problem**
  - **to find the best configurations of a cluster for the whole ETL workflow w.r.t. execution time and monetary cost**
- **Problem modeled as: Multiple Choice Knapsack Problem**
- **Solved by Mixed Integer Linear Programming solver → the lp_solve library (Java)**
  - **impl. at https://github.com/fawadali/MCKPCostModel**
  - S.M.F. Ali, R. Wrembel: Framework to Optimize Data Processing Pipelines Using Performance Metrics. Proc. of DaWaK, LNCS 12393, 2020
  - S.M.F. Ali, R. Wrembel: Towards a Cost Model to Optimize User-Defined Functions in an ETL Workflow Based on User-Defined Performance Metrics. Proc. of ADBIS, LNCS 11695, 2019

ETL optimization with UDFs

- UDFs are frequently treated as black boxes
- Opening a black box
  - discovering semantics
    - code annotations & hints
    - analyzing input / output attributes and values (for simple SQL-based UDFs)
  - discovering performance characteristics
  - discovering patterns in resources' usage → TS analysis



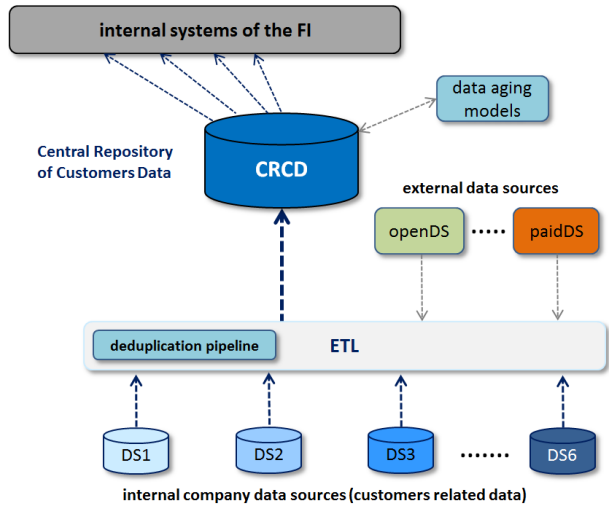
Data processing architecture



- O. Romero, R. Wrembel: Data Engineering for Data Science: Two Sides of the Same Coin. DAWAK, LNCS 12393, 2020

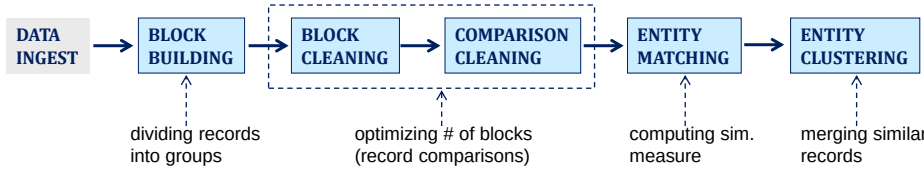
Data integration project @PKO BP

- A project for the biggest Polish bank: PKO BP



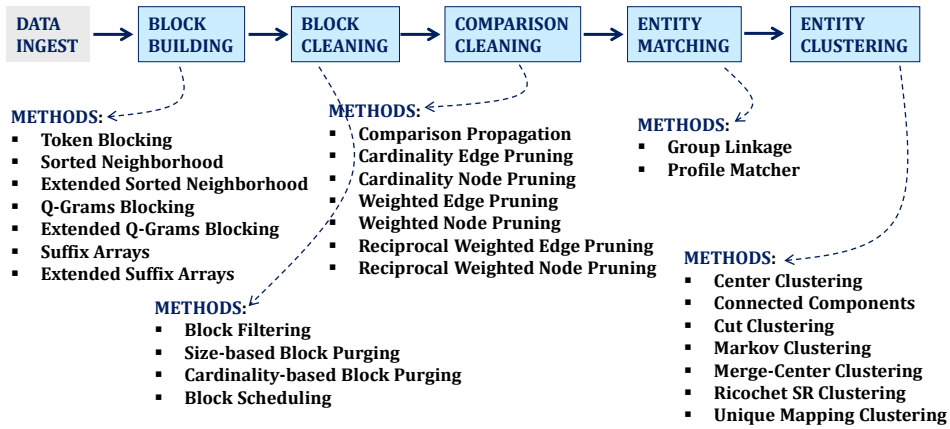
Duplicate data

- Financial institutions
  - from 1% to approx. 5% of duplicate customers' data
- Deduplication complexity: $O(n^2)$
- State of the art deduplication data processing workflow



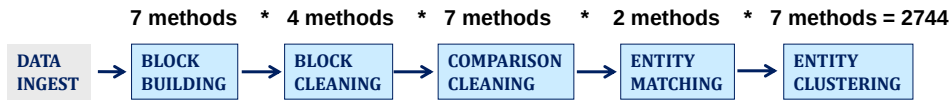
Deduplication

- **Some methods in the deduplication DPW (JedAI)**



- G. Papadakis, L. Tsekouras, E. Thanos, G. Giannakopoulos, T. Palpanas, M. Koubarakis: Domain- and Structure-Agnostic End-to-End Entity Resolution with JedAI. SIGMOD Record, Vol. 48, No. 4, 2019

Deduplication problem



- **Selecting the best combination of methods**
  - **maximizing deduplication quality (precision, recall)**
  - **minimizing execution costs**
  - **greedy approach is impossible**
  - **techniques for pruning search space**
    - **some methods may be better for deduplicating personal data, some for bibliographical data, some for addressess**
    - **knowledge of a domain expert is crucial**
  - **an automatic approach does not exist**

Data aging problem

- Aging data include
  - last names, postal addresses, outdated ID documents, phone numbers, email addressess
- Problem: loosing money for inefficient communication with customers
- Detecting outdated data
  - analog methods applied currently: mailing, phoning, verifying data upon a customer visit → inefficient
- Challenge: to develop models for detecting outdated data automatically

Other topics

- Bitmap index compression
- Indexing dimensions
- Warehousing sequential data
- Data pre-processing for ML
  - with Universitat Politecnica de Catalunya
- Predicting thermal energy consumption
  - for company Kogeneracja Zachód, which builds energy grids in Poland