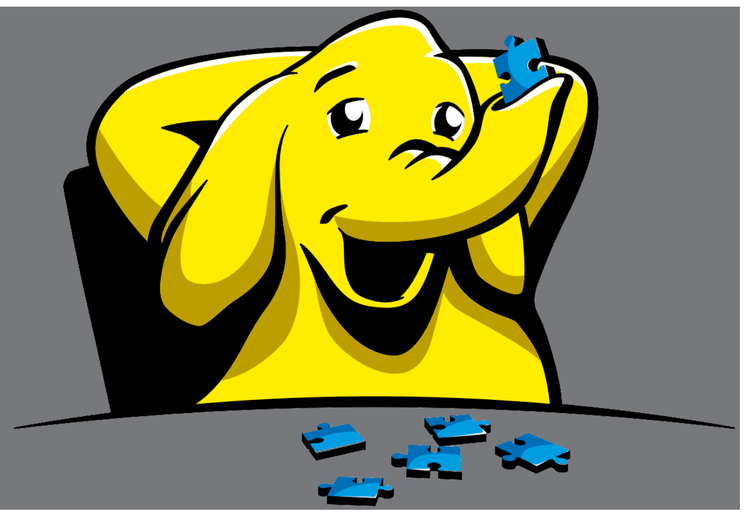
UNIVERSITÄT LEIPZIG

Dedoop
Efficient Deduplication with Hadoop

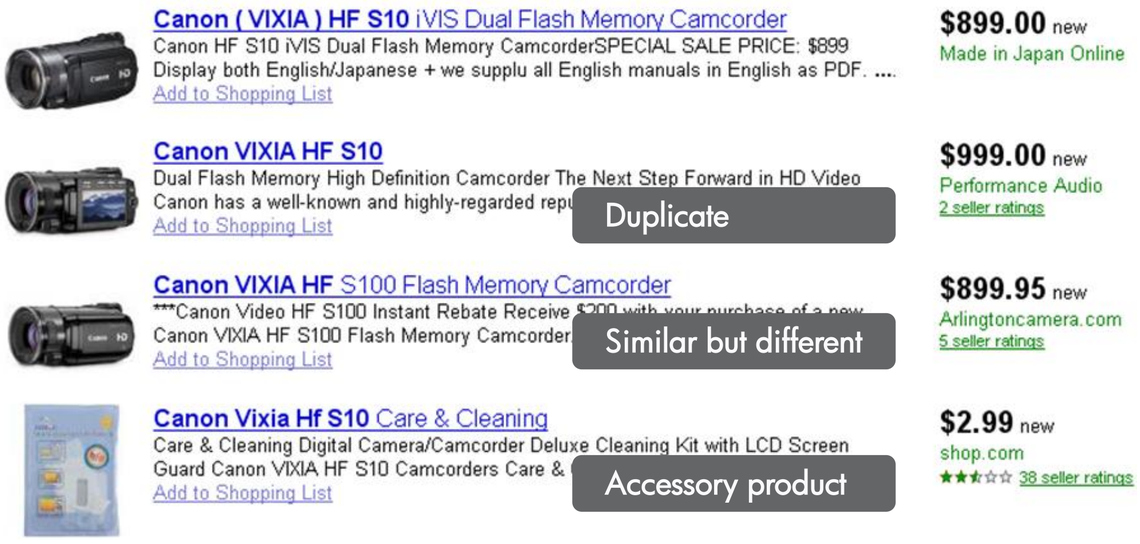
Lars Kolb, Andreas Thor, Erhard Rahm
Database Group, University of Leipzig
http://dbs.uni-leipzig.de



Motivation

What is deduplication?

- Task of identifying duplicates, i.e., entities referring to the same real-world object
- Broad range of applications, e.g.,
  - Duplicate customers in enterprise databases
  - Product offers for price comparison portals



Deduplication is expensive!

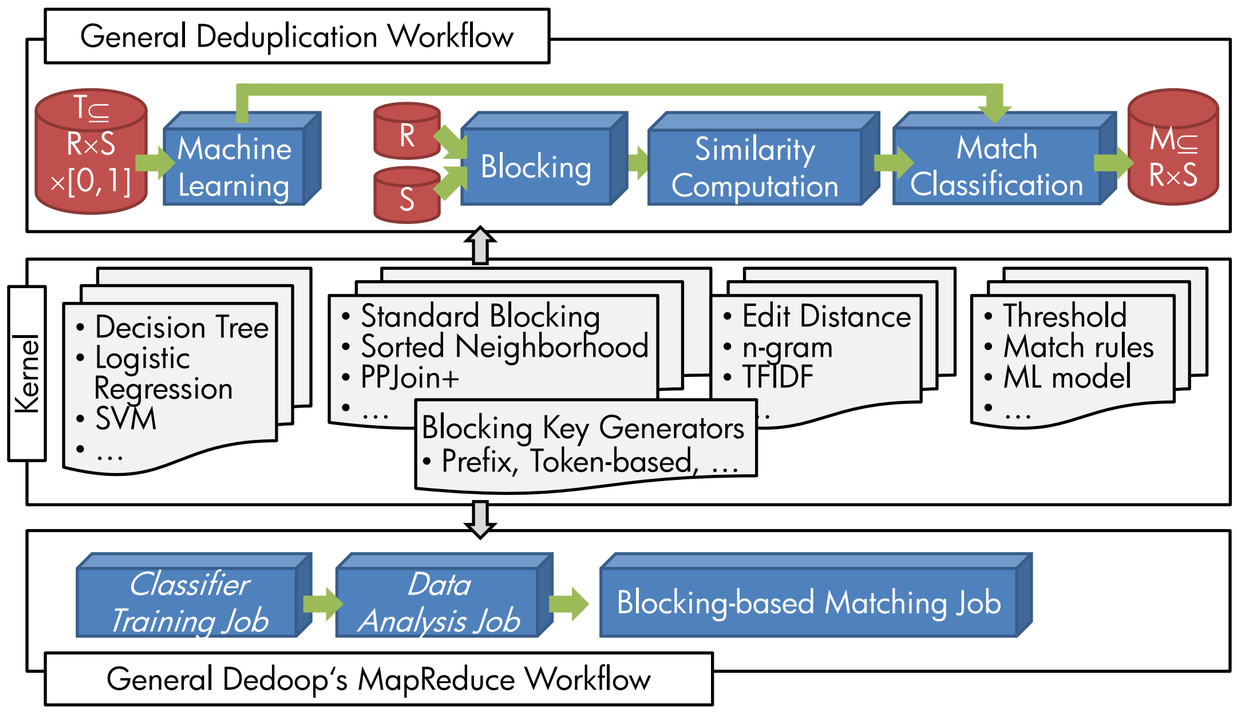
- Pair-wise comparison of input entities
- Application / combination of multiple (domain-specific) similarity measures

→ **Execution on cloud infrastructure**

Dedoop

Deduplication framework based on MapReduce

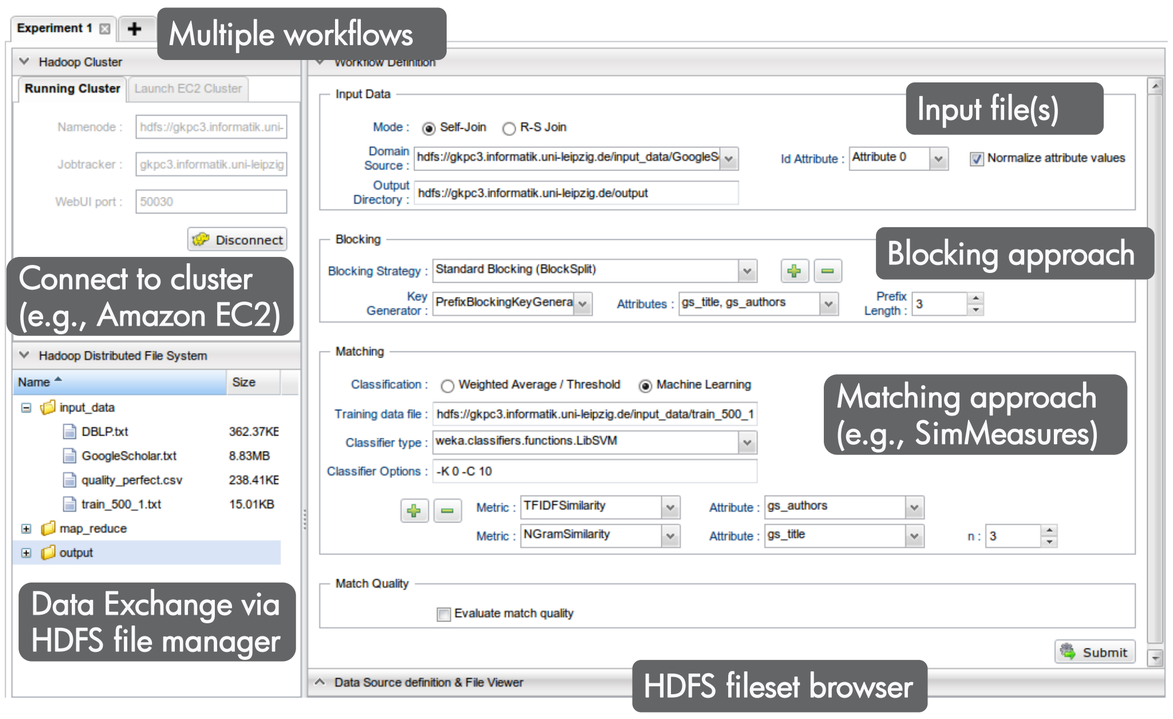
- Browser-based specification of deduplication workflows
- Automatic transformation into executable MapReduce workflows
- Automatic load-balancing
- Automatic elimination of redundant pair-comparisons



Workflow configuration

Browser-based specification of advanced deduplication workflows

- Rich toolset of common blocking techniques and similarity functions
- Support of machine learning-based match classifiers



Workflow management

- Automatic mapping of specified workflow(s) to a sequence of MR jobs
- Automatic workflow submission to Hadoop cluster incl. progress monitoring
- Simultaneous handling of multiple workflows and multiple clusters
- Convenient cluster management (e.g., automatic launching of Amazon EC2 VMs)

Efficient cluster utilization

Basic blocking approach with MapReduce

- Map – determine blocking key for every input entity and output (blockkey, entity) pair
- Partitioning by blocking key and block-wise redistribution to *r* reduce tasks
- Reduce – matching of entities of the same block

1. Load imbalances

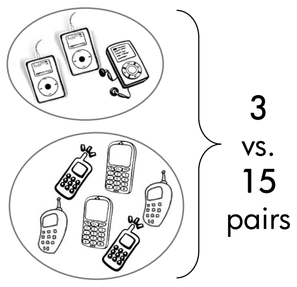
- Susceptible to severe load imbalances due to skewed block sizes
- Execution time dominated by a few tasks that process the largest blocks

→ Dedoop: Automatic techniques for balancing workload across all reduce tasks

Example: 2 reduce tasks, blocking 9 products by type: 3 MP3 players vs. 6 cell phones

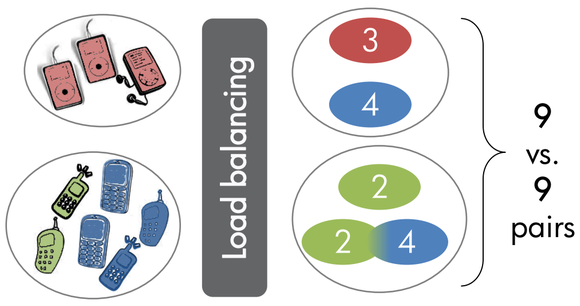
Basic approach

- Simple block distribution (1 block per task)



BlockSplit approach

- Data Analysis Job: Split large blocks into sub-blocks
- Block distribution based on size of sub-blocks

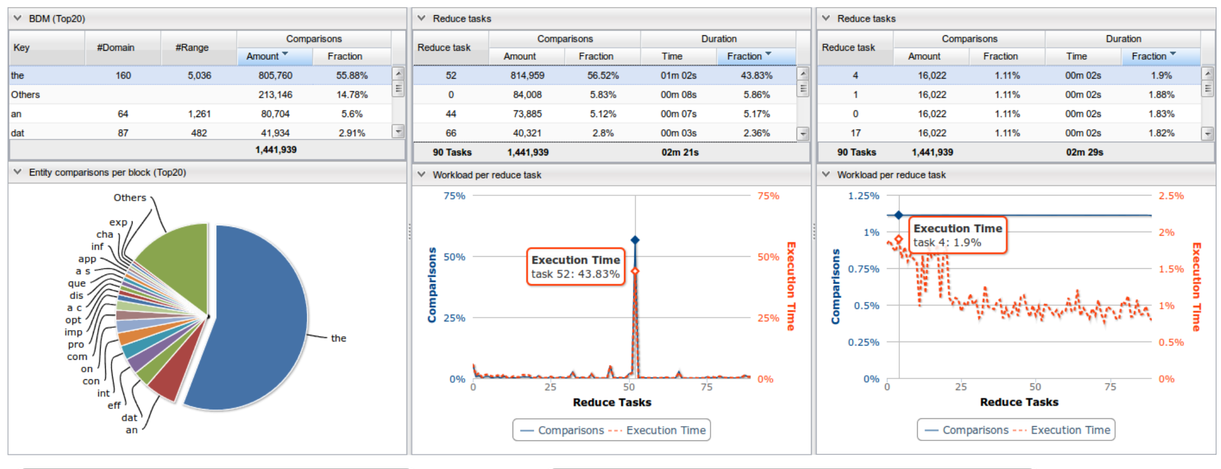


2. Redundant pair-comparisons

- Entities can be assigned to multiple blocks (e.g., multi-phase blocking)
- Parallel execution may lead to unnecessary comparisons (same pair in multiple blocks)

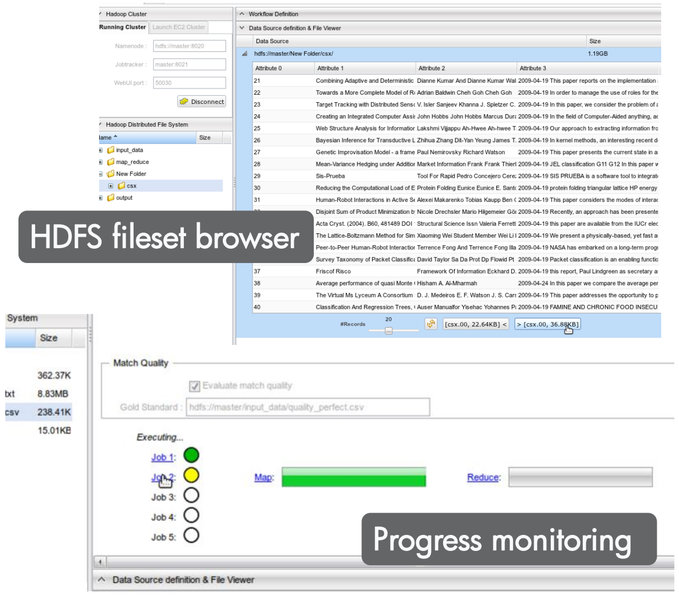
→ Dedoop: Automatic techniques for eliminating redundant pair comparisons

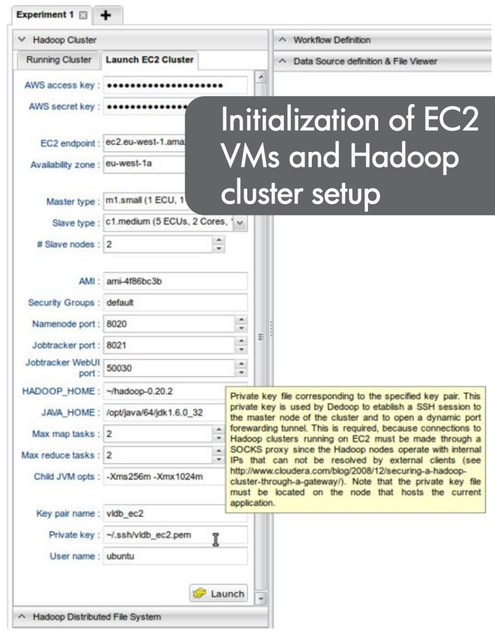
Screenshots



Data skew, i.e.,unbalanced distribution of block sizes

Workload of reduce tasks with (left) and without load balancing (right)





Printed by Universitätsrechenzentrum Leipzig