Load Balancing for MapReduce-based Entity Resolution

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

Motivation

Entity Resolution

- Task of identifying entities referring to the same real-world object
- Application of similarity measures on pairs of input entities
  - Evaluation of Cartesian product leads to complexity of \( O(n^2) \)
  - Based on entity signatures (blocking keys), blocking techniques semantically group similar entities in blocks and restrict matching to entities of the same block

Basic approach

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

Goals

- Parallelization of time-intensive Blocking-based Entity Resolution with MapReduce
- Load balancing mechanism to evenly utilize available compute capacity ensuring effectiveness and scalability

Load Balancing – Overview

Example without Load Balancing

Basic approach (m=2 input partitions/map tasks, r=3 reduce tasks)

Problem

- Susceptible to severe load imbalances due to skewed block sizes
- Execution time dominated by a few tasks that process the largest block
- Large blocks prevent utilization of more than a few nodes

Example with Load Balancing (BlockSplit)

Analysis job

- Average workload per reduce task = \( 20/3 \approx 6.7 \)
- Large block size \( (PF=10 > 6.7) \) split in \( m=2 \) subblocks
- \( \Phi_{m+1} \rightarrow \Phi_m \times \Phi_j \times \text{match tasks} \times 3.0 \times 3.1 \times 3.0 \)

Match job

- Composite keys – reduceTask.block.split
- Replication of entities by map
- part(reduceTask.block.split)=reduceTask

Related work

- L. Kolb, A. Thor, and E. Rahm. Parallel Sorted Neighborhood Blocking with MapReduce. BTW, 2011
- L. Kolb, A. Thor, and E. Rahm. Multi-pass Sorted Neighborhood Blocking with MapReduce. CSRD 27(1), 2012

Experimental Results (n=# dual core VMs, m=# map tasks, r=# reduce tasks)

Robustness against data skew

- 100 blocks, size of \( \Phi \) block is proportional to \( n^{-2} \)
- 114,000 entities, \( n=10, m=20, r=100 \)

Scalability

- 114,000 entities \( \rightarrow 3 \times 10^8 \) comparisons
- \( n \in [1, 100], m=2, n=10 \cdot n \)