**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)**

<table>
<thead>
<tr>
<th>Input Entities</th>
<th>MR Job: Analysis</th>
<th>MR Job: Blocking</th>
<th>Merge with</th>
<th>Reduce tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1, T2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1, T2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1, T2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1, T2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Example with Load Balancing (BlockSplit)**

**Analysis job**
- Average workload per reduce task $= \frac{20}{3} \approx 6.6$
- Large block $\Phi_1, \Phi_2 \succ \Phi$ (FP=10 $\times 6.6$) split in $m=2$ sub-blocks

**Match job**
- Composite keys - reduceTask block.split
- Replication of entities by map
- parallelTask block.split= reduceTask

**Robustness against data skew**
- 100 blocks – size of $k^2$ block is proportional to $e^{k^2}$
- 114,000 entities, $m=10$, $n=20$, $r=100$

**Scaleability**
- 114,000 entities $\rightarrow 3\cdot10^6$ comparisons
- $n \in [1,100]$, $m=2$, $n=10^n$

**Related work**
- L. Kolb, A. Thor, and E. Rahm. Parallel Sorted Neighborhood Blocking with MapReduce. BTW, 2011