Load Balancing for MapReduce-based Entity Resolution

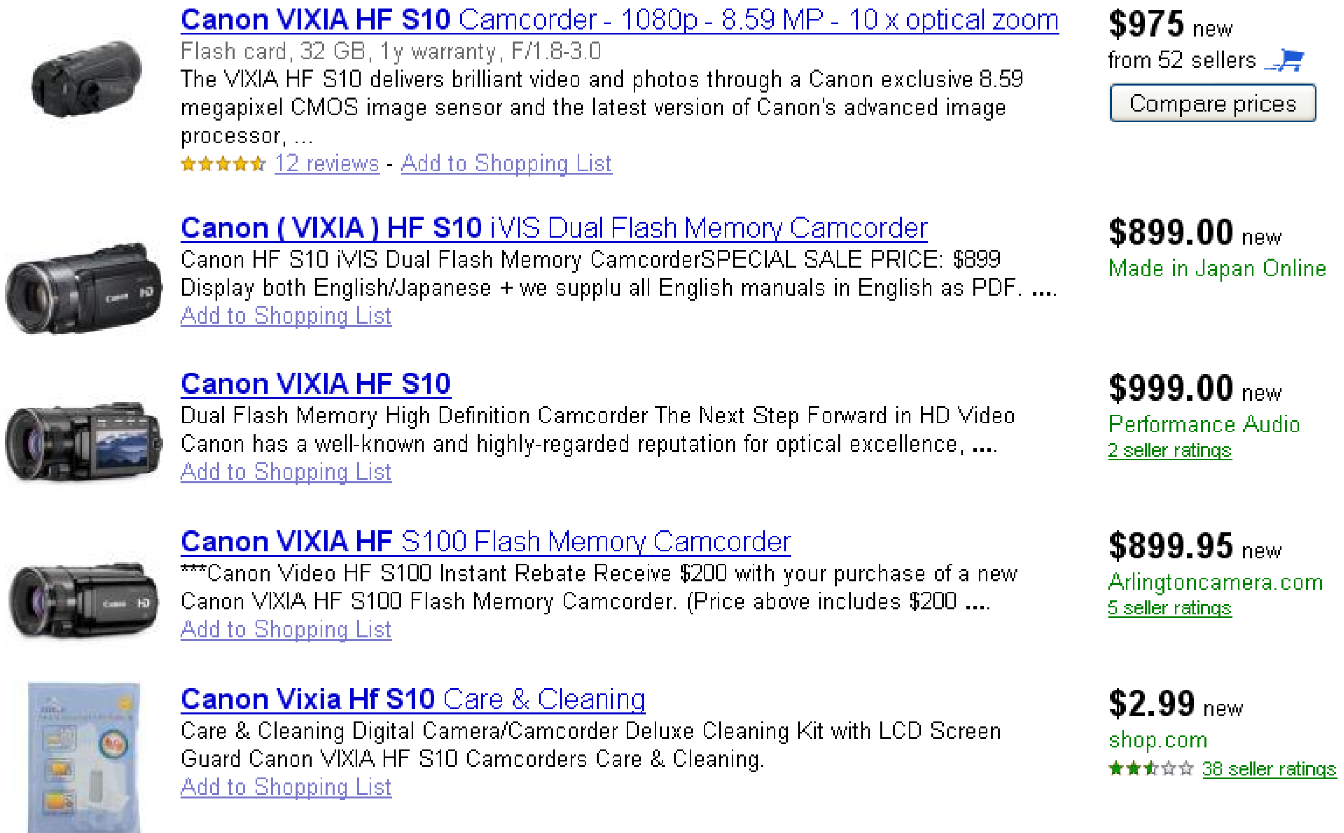
Lars Kolb, Andreas Thor, Erhard Rahm
University of Leipzig

April 3, 2012

http://dbs.uni-leipzig.de

Entity Resolution

- Identification of semantically equivalent entities (objects)
  - within one source or between two sources
  - to merge them, compare them, improve data quality, etc.



Canon VIXIA HF S10 Camcorder - 1080p - 8.59 MP - 10 x optical zoom
Flash card, 32 GB, 1y warranty, F/1.8-3.0
The VIXIA HF S10 delivers brilliant video and photos through a Canon exclusive 8.59 megapixel CMOS image sensor and the latest version of Canon's advanced image processor, ...
★★★★★ 12 reviews - Add to Shopping List

$975 new
from 52 sellers
Compare prices

Canon (VIXIA) HF S10 iVIS Dual Flash Memory Camcorder
Canon HF S10 iVIS Dual Flash Memory CamcorderSPECIAL SALE PRICE: $899 Display both English/Japanese + we supplu all English manuals in English as PDF. ....
Add to Shopping List

$899.00 new
Made in Japan Online

Canon VIXIA HF S10
Dual Flash Memory High Definition Camcorder The Next Step Forward in HD Video Canon has a well-known and highly-regarded reputation for optical excellence, ...
Add to Shopping List

$999.00 new
Performance Audio
2 seller ratings

Canon VIXIA HF S100 Flash Memory Camcorder
***Canon Video HF S100 Instant Rebate Receive $200 with your purchase of a new Canon VIXIA HF S100 Flash Memory Camcorder. (Price above includes $200 ....
Add to Shopping List

$899.95 new
Arlingtoncamera.com
5 seller ratings

Canon Vixia Hf S10 Care & Cleaning
Care & Cleaning Digital Camera/Camcorder Deluxe Cleaning Kit with LCD Screen Guard Canon VIXIA HF S10 Camcorders Care & Cleaning.
Add to Shopping List

$2.99 new
shop.com
★★★☆☆ 38 seller ratings

Entity Resolution Problem

- Lot of research work
  - String similarities, usage of structural information
  - Combined use of several matching approaches
  - Application of machine learning
  - …
- Study of real-world match systems/problems [VLDB'10]
  - Effective entity resolution is difficult: F-Measure <75% for product data
  - Entity resolution is expensive: scalability issues for O(n^2)

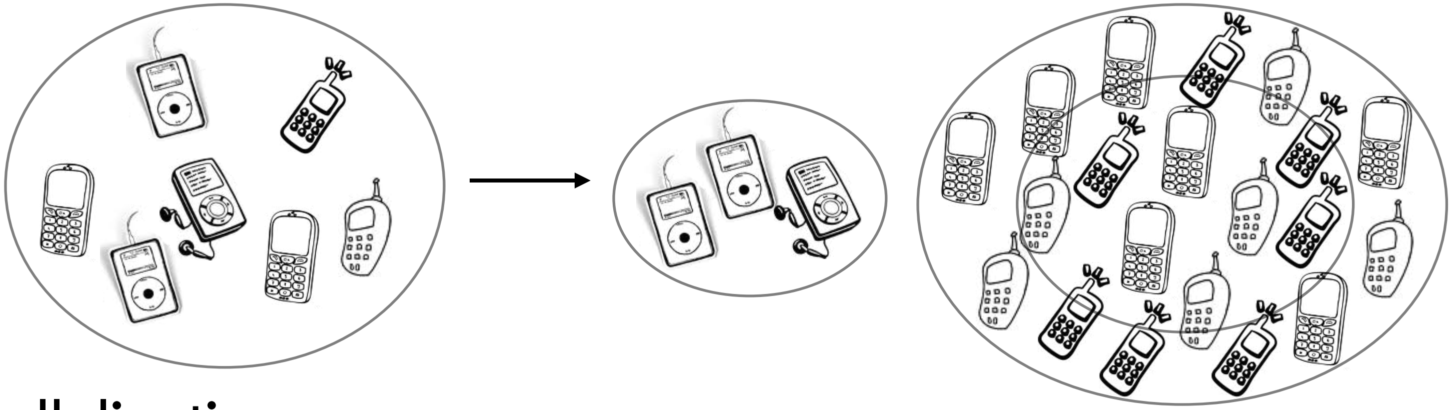
[VLDB'10] Koepcke, Thor, Rahm: Evaluation of entity resolution approaches on real-world match problems. VLDB 2010

Outline

- Entity Resolution
- Blocking-based Entity Resolution with MapReduce
- Load Balancing
  - Problem
  - Block-Split Approach
- Experimental Results
- Conclusions & Future Work

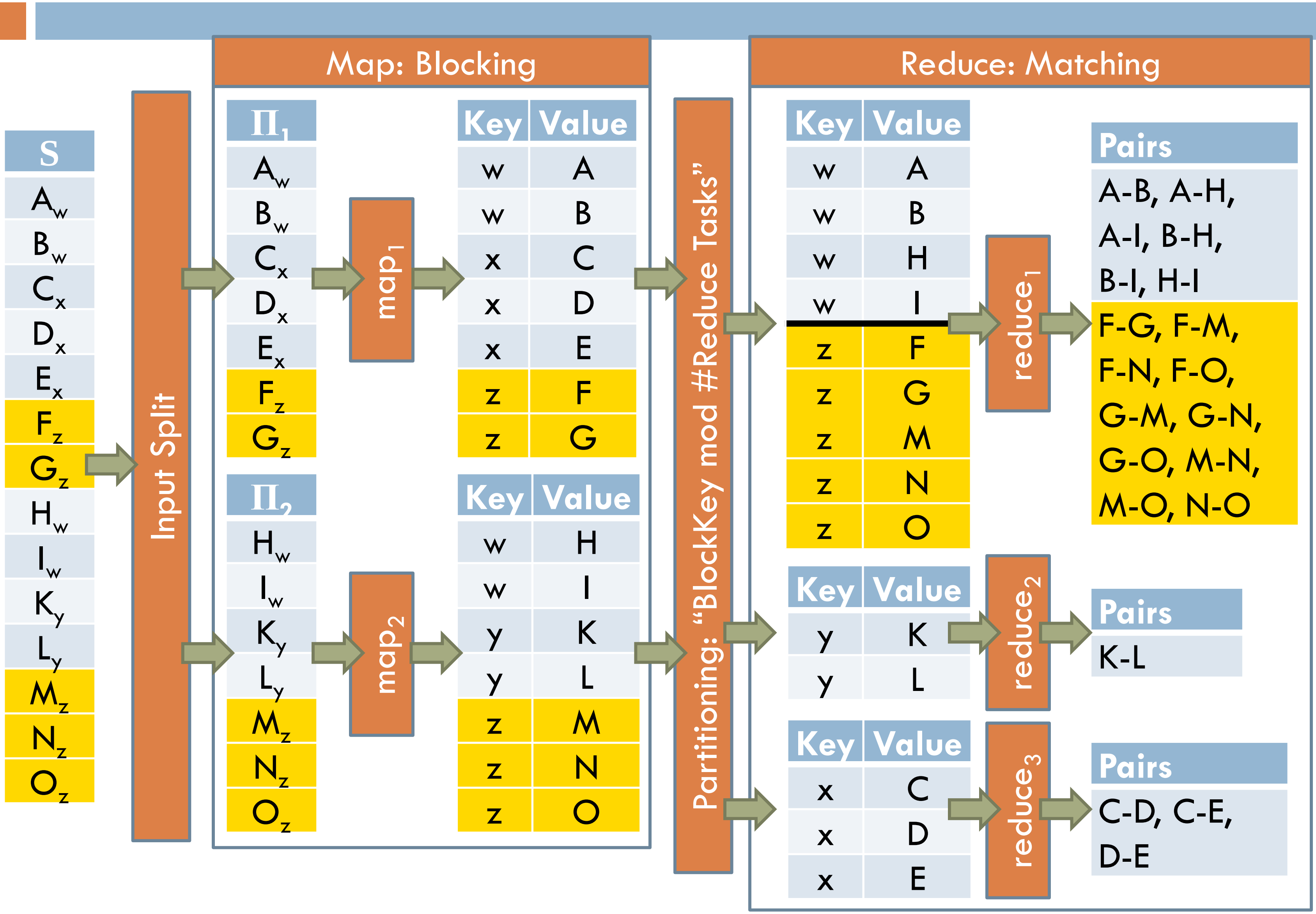
How to speed up entity matching?

- Entity matching is expensive (due to pair-wise comparisons)
- **Blocking** to reduce search space
  - Group similar entities within blocks based on blocking key
  - Restrict matching to entities from the same block



- Parallelization
  - Split match computation in sub-tasks to be executed in parallel
  - Exploitation of cloud infrastructures and frameworks like MapReduce

Blocking + MapReduce: Naïve



S
A_w
B_w
C_x
D_x
E_x
F_z
G_z
H_w
I_w
K_y
L_y
M_z
N_z
O_z

Map: Blocking

Π_1
A_w
B_w
C_x
D_x
E_x
F_z
G_z

Key	Value
w	A
w	B
x	C
x	D
x	E
z	F
z	G

Π_2
H_w
I_w
K_y
L_y
M_z
N_z
O_z

Key	Value
w	H
w	I
y	K
y	L
z	M
z	N
z	O

Reduce: Matching

Key	Value
w	A
w	B
w	H
w	I
z	F
z	G
z	M
z	N
z	O

Pairs
A-B, A-H, A-I, B-H, B-I, H-I
F-G, F-M, F-N, F-O, G-M, G-N, G-O, M-N, M-O, N-O

Key	Value
y	K
y	L

Pairs
K-L

Key	Value
x	C
x	D
x	E

Pairs
C-D, C-E, D-E

Load Balancing: Problem

- Data skew leads to unbalanced workload
  - Large blocks prevent utilization of more than a few nodes
  - Deteriorates scalability and efficiency
  - Unnecessary costs (you also pay for underutilized machines!)
- **Key ideas** for load balancing
  - Additional MR job to determine blocking key distribution, i.e., number and size of blocks (per input partition)
  - Global load balancing that assigns (nearly) the same number of pairs to reduce tasks

Load Balancing: Approaches

- Two load balancing strategies for parallel entity resolution with general blocking

- **BlockSplit**: Split large blocks into sub-blocks
- **PairRange**: Global enumeration and tailored distribution of all pairs

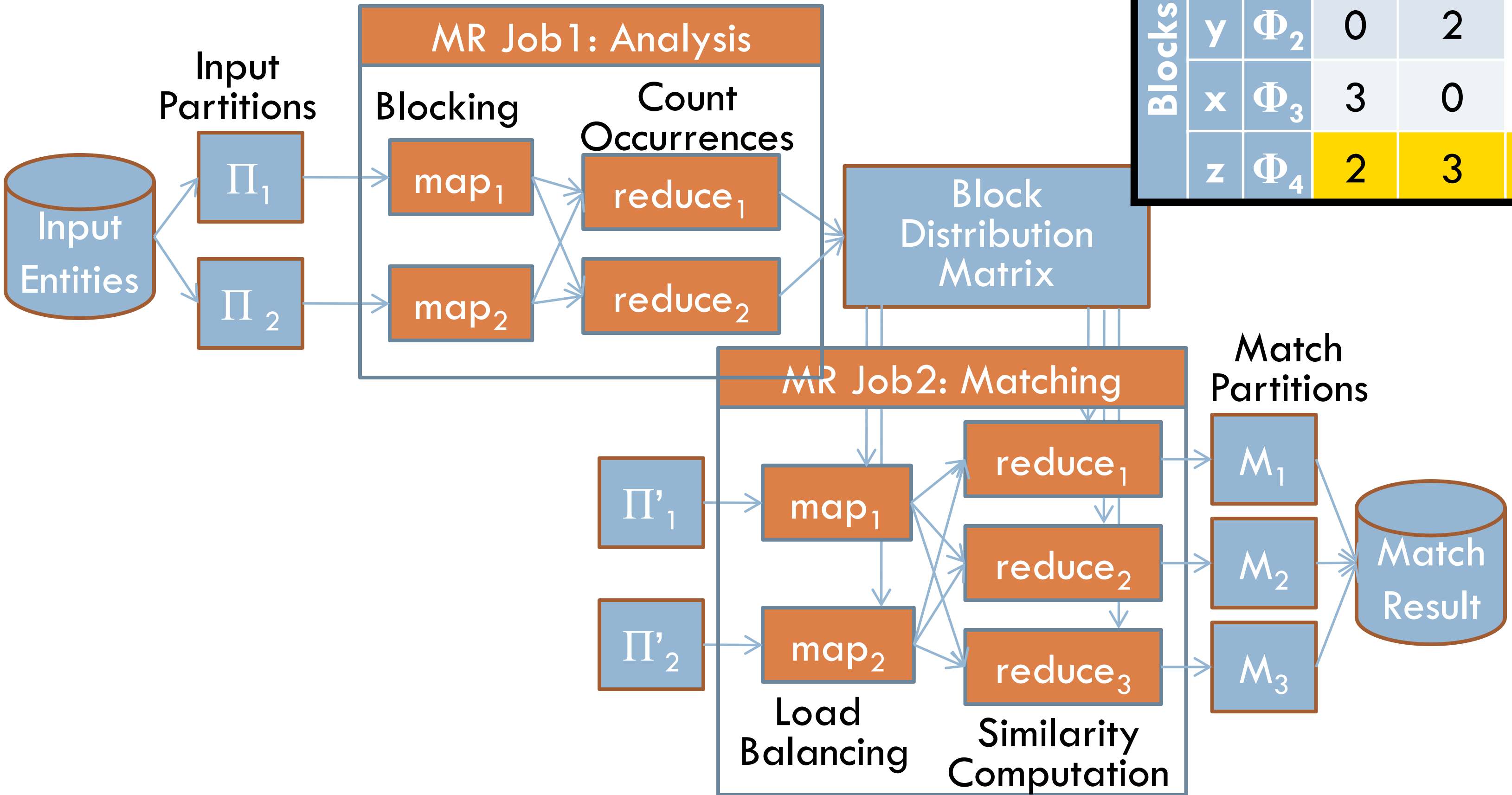
- Variation for Sorted Neighborhood [CSRD'12]

[CSRD'12] Kolb, Thor, Rahm: Multi-pass Sorted Neighborhood Blocking with MapReduce. Computer Science - Research and Development 27(1), 2012

Load Balancing for MR-based Entity Res.

Partition	Π_1							Π_2						
Entity	A	B	C	D	E	F	G	H	I	K	L	M	N	O
Blocking Key	w	w	x	x	x	z	z	w	w	y	y	z	z	z

			Partition		Overall	
			Π_1	Π_2	#E	#P
Blocks	w	Φ_1	2	2	4	6
	y	Φ_2	0	2	2	1
	x	Φ_3	3	0	3	3
	z	Φ_4	2	3	5	10



BlockSplit

- Large blocks split into *m* sub-blocks
  - according to *m* input partitions
  - large if $\#P_{Block} > \#P_{Overall} / \#Reducer$
- Two types of match tasks
  - Single (small blocks and sub-blocks)
  - Two sub-blocks
- Greedy load balancing
  - Sort match tasks by number of pairs in descending order
  - Assign match task to reducer with lowest number of pairs
- **Example**
  - r=3 reduce tasks, split Φ_4 in m=2 sub-blocks
  - Φ_4's match tasks: $\Phi_{4.1}$, $\Phi_{4.2}$, and $\Phi_{4.1\times2}$

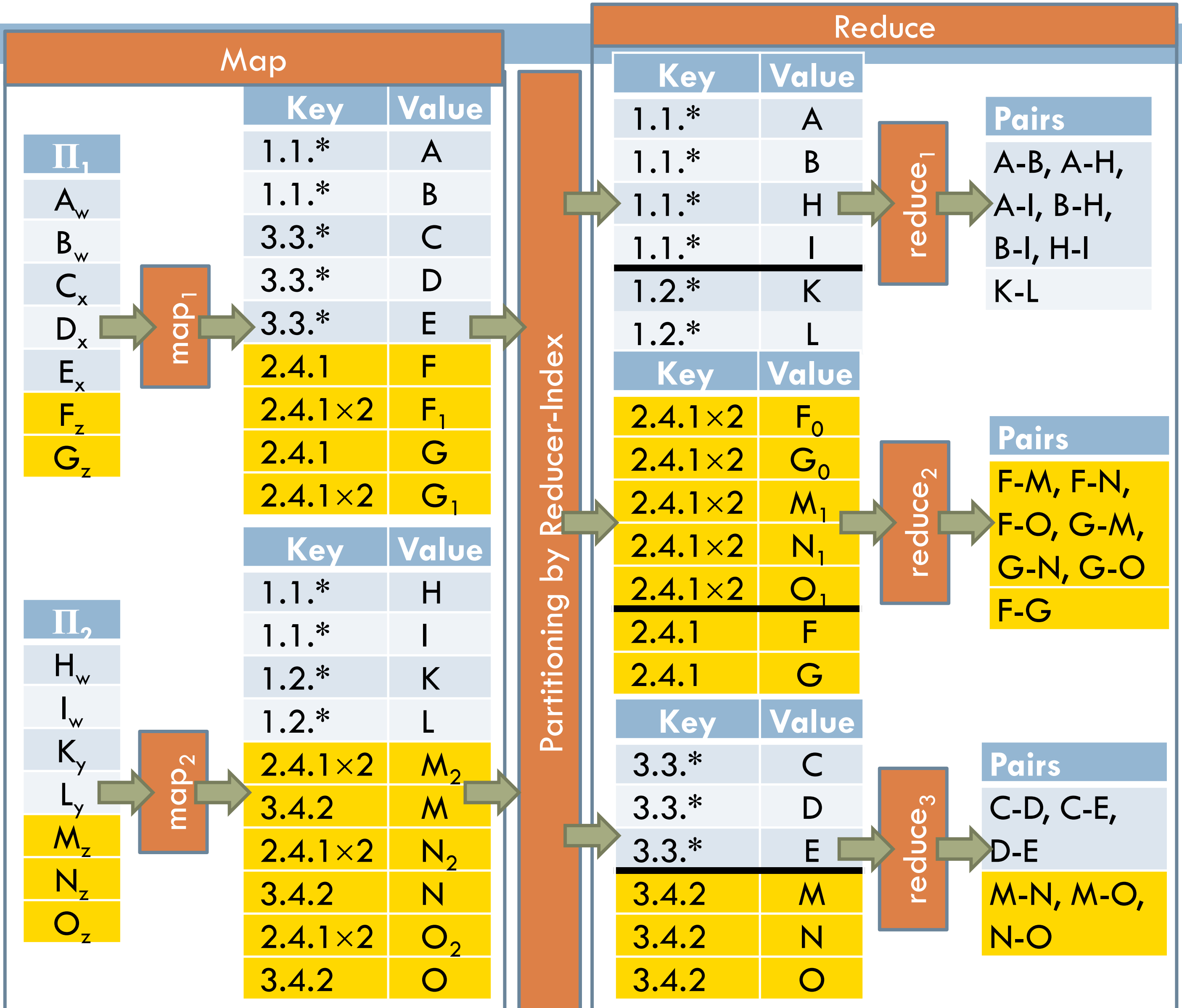
			Partition		Overall	
			Π_1	Π_2	#E	#P
Blocks	w	Φ_1	2	2	4	6
	y	Φ_2	0	2	2	1
	x	Φ_3	3	0	3	3
	z	Φ_4	2	3	5	10

		#P	Reducer
Match Tasks	Φ_1	6	1
	$\Phi_{4.1\times2}$	6	2
	Φ_3	3	3
	$\Phi_{4.2}$	3	3
	Φ_2	1	1
	$\Phi_{4.1}$	1	2

BlockSplit: MapReduce Dataflow

MapReduce Techniques

- MapKey = ReducerIndex + MatchTask
- Replicate entities of sub-blocks



Π_1
A_w
B_w
C_x
D_x
E_x
F_z
G_z

Key	Value
1.1.*	A
1.1.*	B
3.3.*	C
3.3.*	D
3.3.*	E
2.4.1	F
2.4.1×2	F_1
2.4.1	G
2.4.1×2	G_1

Π_2
H_w
I_w
K_y
L_y
M_z
N_z
O_z

Key	Value
1.1.*	H
1.1.*	I
1.2.*	K
1.2.*	L
2.4.1×2	M_2
3.4.2	M
2.4.1×2	N_2
3.4.2	N
2.4.1×2	O_2
3.4.2	O

Key	Value
1.1.*	A
1.1.*	B
1.1.*	H
1.1.*	I
1.2.*	K
1.2.*	L

Pairs
A-B, A-H, A-I, B-H, B-I, H-I
K-L

Key	Value
2.4.1×2	F_0
2.4.1×2	G_0
2.4.1×2	M_1
2.4.1×2	N_1
2.4.1×2	O_1
2.4.1	F
2.4.1	G

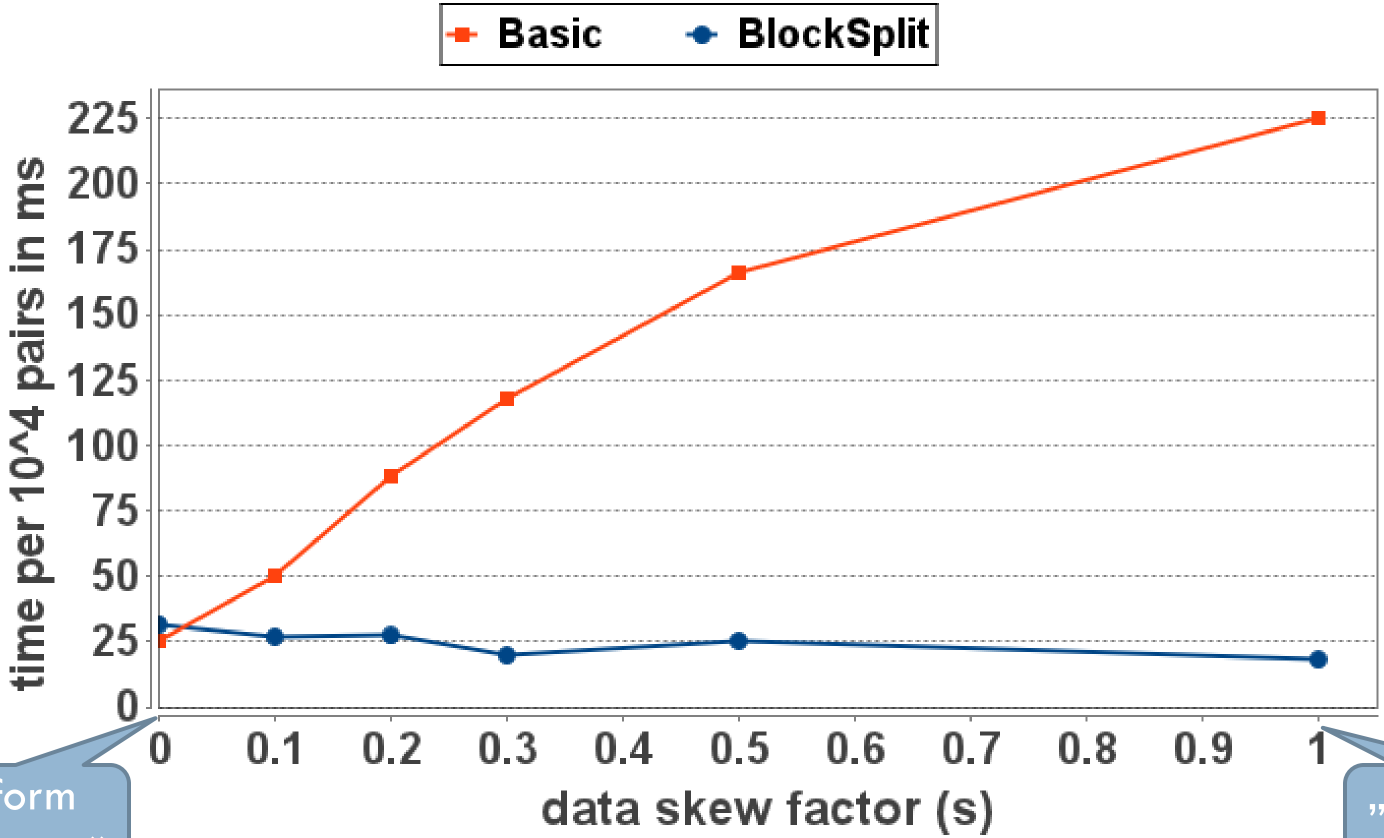
Pairs
F-M, F-N, F-O, G-M, G-N, G-O
F-G

Key	Value
3.3.*	C
3.3.*	D
3.3.*	E
3.4.2	M
3.4.2	N
3.4.2	O

Pairs
C-D, C-E, D-E
M-N, M-O, N-O

Evaluation: Data Skew

- BlockSplit **robust** against data skew
  - Evaluation on Amazon EC2; 114.000 product records

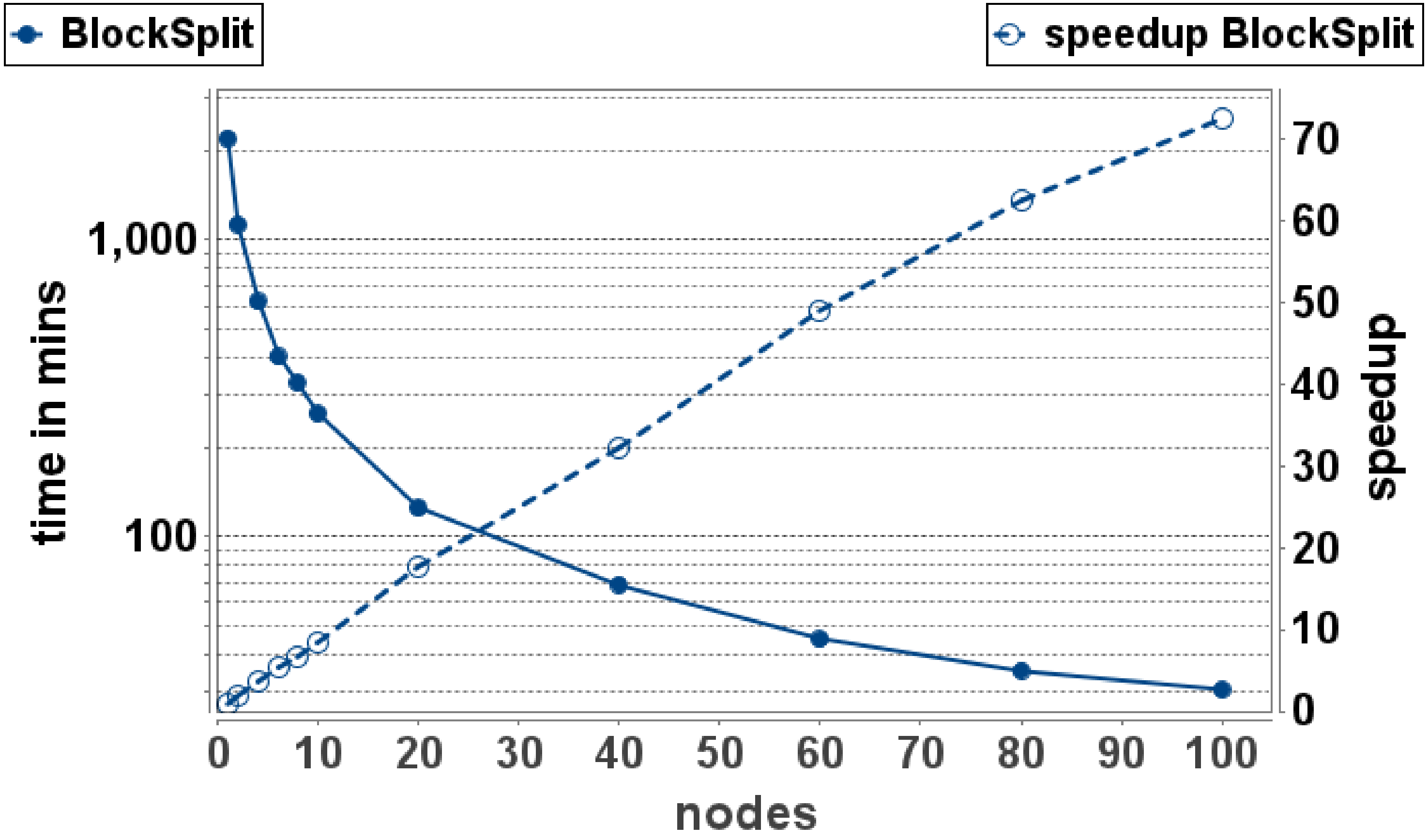


„Uniform distribution“

„All entities in a single block“

Evaluation: Scalability

- BlockSplit is **scalable**



Conclusions and Future Work

- Faster entity resolution by
  - Blocking
  - Parallel matching
- Straight-forward utilization of MapReduce possible
  - … but doing it efficiently requires some work
- Effective load balancing approaches such as Block-Split
  - Additional MR job for analysis incurs minimal overhead

- Future Work
  - Load balancing for other data-intensive tasks
  - Analytic model for determining #reduce tasks

Thank you!