



Entity Resolution with Iterative Blocking

Steven Whang, David Menestrina, Georgia Koutrika,
Martin Theobald, Hector Garcia-Molina

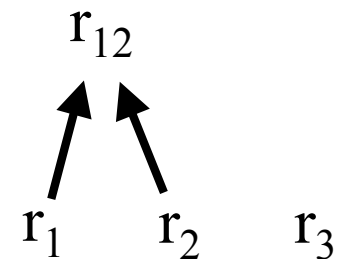
Stanford University

Entity Resolution

Record	Name	Zip	Email
r_1	John Doe	12345	jdoe@yahoo
r_2	John Doe	94305	
r_3	J. Foe	94305	jdoe@yahoo

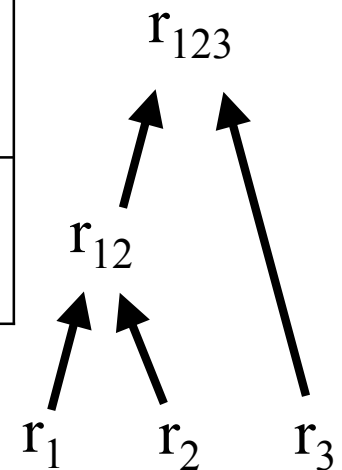
Entity Resolution

Record	Name	Zip	Email
r_1	John Doe	12345	jdoe@yahoo
r_2	John Doe	94305	
r_3	J. Foe	94305	jdoe@yahoo
r_{12}	John Doe	{12345, 94305}	jdoe@yahoo



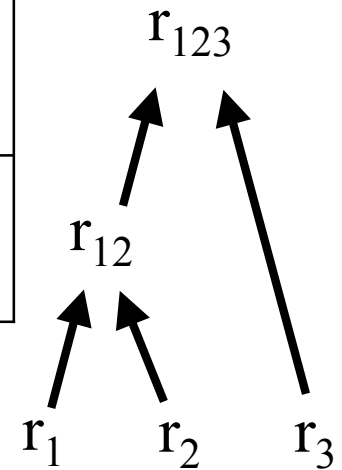
Entity Resolution

Record	Name	Zip	Email
r_1	John Doe	12345	jdoe@yahoo
r_2	John Doe	94305	
r_3	J. Foe	94305	jdoe@yahoo
r_{12}	John Doe	{12345, 94305}	jdoe@yahoo
r_{123}	{John Doe, J. Foe}	{12345, 94305}	jdoe@yahoo



Entity Resolution

Record	Name	Zip	Email
r_1	John Doe	12345	jdoe@yahoo
r_2	John Doe	94305	
r_3	J. Foe	94305	jdoe@yahoo
r_{12}	John Doe	{12345, 94305}	jdoe@yahoo
r_{123}	{John Doe, J. Foe}	{12345, 94305}	jdoe@yahoo

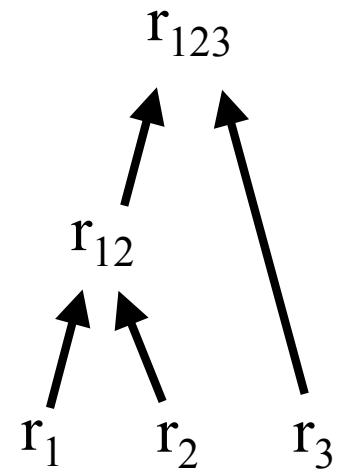


Exhaustive Solution: $\{r_{123}\}$

Blocking

Record	Name	Zip	Email
r_1	John Doe	12345	jdoe@yahoo
r_2	John Doe	94305	
r_3	J. Foe	94305	jdoe@yahoo

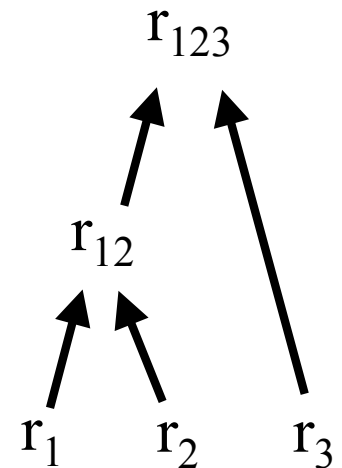
Partition by	Block#1	Block#2
zip	r_1	r_2, r_3



Blocking

Record	Name	Zip	Email
r_1	John Doe	12345	jdoe@yahoo
r_2	John Doe	94305	
r_3	J. Foe	94305	jdoe@yahoo

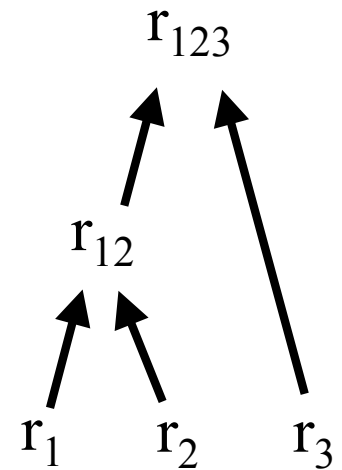
Partition by	Block#1	Block#2
zip	r_1	r_2, r_3
1 st char last name	r_1, r_2	r_3



Blocking

Record	Name	Zip	Email
r_1	John Doe	12345	jdoe@yahoo
r_2	John Doe	94305	
r_3	J. Foe	94305	jdoe@yahoo

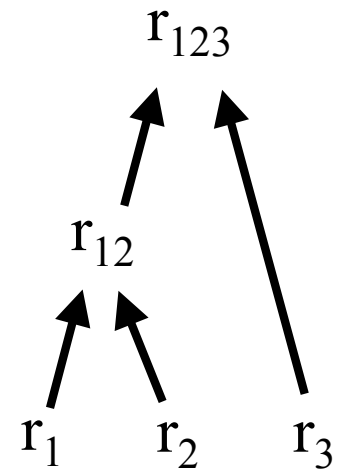
Partition by	Block#1	Block#2
zip	r_1	r_2, r_3
1 st char last name	r_{12}	r_3



Blocking

Record	Name	Zip	Email
r_1	John Doe	12345	jdoe@yahoo
r_2	John Doe	94305	
r_3	J. Foe	94305	jdoe@yahoo

Partition by	Block#1	Block#2
zip	r_1	r_2, r_3
1 st char last name	r_{12}	r_3

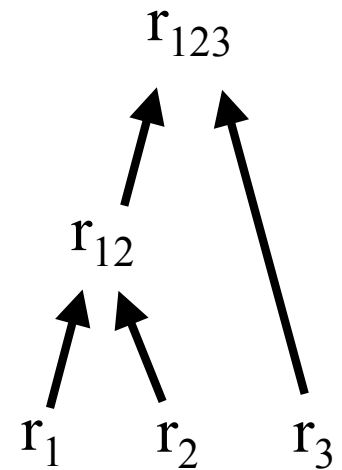


Blocking Solution: $\{r_{12}, r_3\}$

Iterative Blocking

Record	Name	Zip	Email
r_1	John Doe	12345	jdoe@yahoo
r_2	John Doe	94305	
r_3	J. Foe	94305	jdoe@yahoo

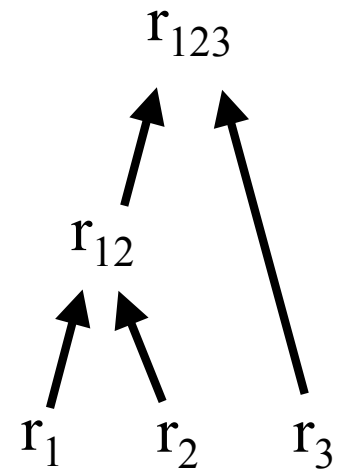
Partition by	Block#1	Block#2
zip	r_1	r_2, r_3
1 st char last name	r_1, r_2	r_3



Iterative Blocking

Record	Name	Zip	Email
r_1	John Doe	12345	jdoe@yahoo
r_2	John Doe	94305	
r_3	J. Foe	94305	jdoe@yahoo

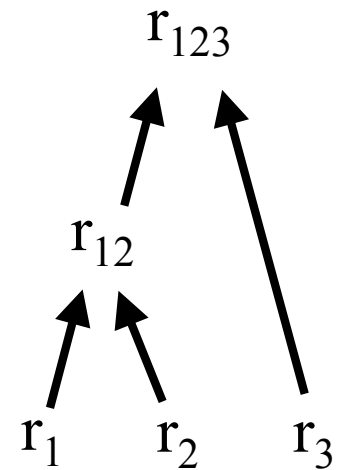
Partition by	Block#1	Block#2
zip	r_1	r_2, r_3
1 st char last name	r_{12}	r_3



Iterative Blocking

Record	Name	Zip	Email
r_1	John Doe	12345	jdoe@yahoo
r_2	John Doe	94305	
r_3	J. Foe	94305	jdoe@yahoo

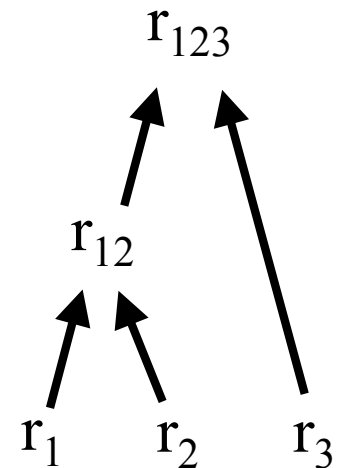
Partition by	Block#1	Block#2
zip	r_1, r_{12}	r_2, r_3, r_{12}
1 st char last name	r_{12}	r_3



Iterative Blocking

Record	Name	Zip	Email
r_1	John Doe	12345	jdoe@yahoo
r_2	John Doe	94305	
r_3	J. Foe	94305	jdoe@yahoo

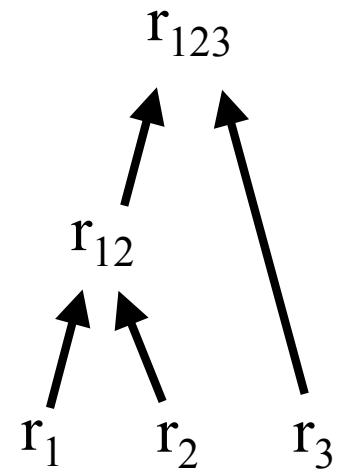
Partition by	Block#1	Block#2
zip	r_{12}	r_{123}
1 st char last name	r_{12}	r_3



Iterative Blocking

Record	Name	Zip	Email
r_1	John Doe	12345	jdoe@yahoo
r_2	John Doe	94305	
r_3	J. Foe	94305	jdoe@yahoo

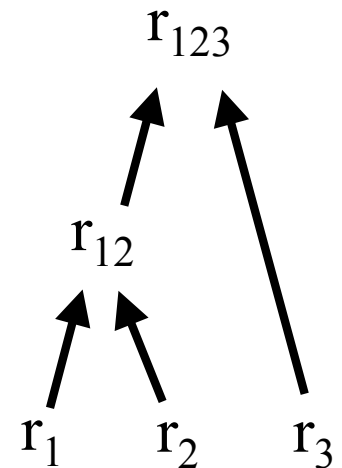
Partition by	Block#1	Block#2
zip	r_{123}	r_{123}
1 st char last name	r_{123}	r_{123}



Iterative Blocking

Record	Name	Zip	Email
r_1	John Doe	12345	jdoe@yahoo
r_2	John Doe	94305	
r_3	J. Foe	94305	jdoe@yahoo

Partition by	Block#1	Block#2
zip	r_{123}	r_{123}
1 st char last name	r_{123}	r_{123}



Iterative Blocking Solution: $\{r_{123}\}$

Overview

- Model
- Algorithms
- Experimental Results

Iterative Blocking Model

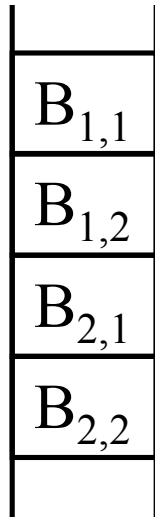
- Result is the fixed-point state of applying a “core” ER algorithm on blocks and re-distributing new records
- Can plug in any core ER algorithm that partitions records

In-memory Algorithm (Lego)

- Maintains “maximal records” for efficient block updates
- Maintains a queue of blocks to process

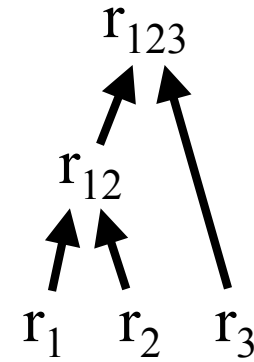
Example

$B_{1,1}$ r_1	$B_{1,2}$ r_2, r_3
$B_{2,1}$ r_1, r_2	$B_{2,2}$ r_3



Block Queue

Current Block



$$r_1 \Rightarrow r_1$$

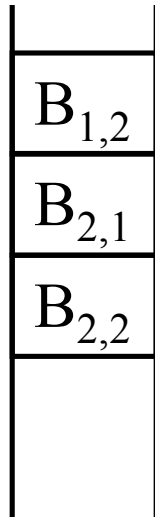
$$r_2 \Rightarrow r_2$$

$$r_3 \Rightarrow r_3$$

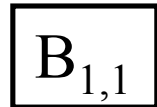
Maximal Records

Example

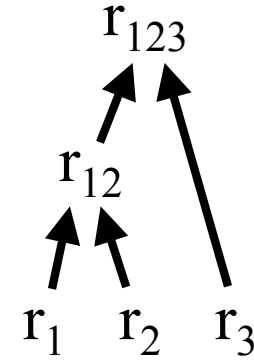
$B_{1,1}$ r_1	$B_{1,2}$ r_2, r_3
$B_{2,1}$ r_1, r_2	$B_{2,2}$ r_3



Block Queue



Current Block



$$r_1 \Rightarrow r_1$$

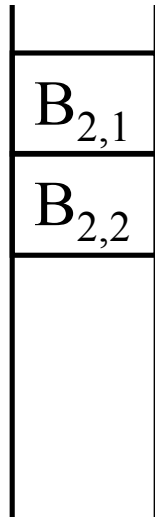
$$r_2 \Rightarrow r_2$$

$$r_3 \Rightarrow r_3$$

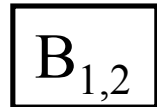
Maximal Records

Example

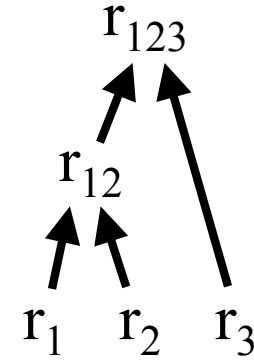
$B_{1,1}$ r_1	$B_{1,2}$ r_2, r_3
$B_{2,1}$ r_1, r_2	$B_{2,2}$ r_3



Block Queue



Current Block



$$r_1 \Rightarrow r_1$$

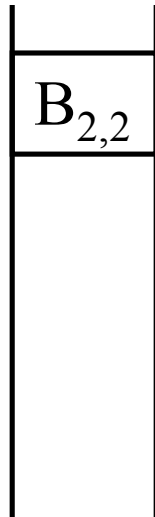
$$r_2 \Rightarrow r_2$$

$$r_3 \Rightarrow r_3$$

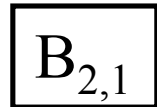
Maximal Records

Example

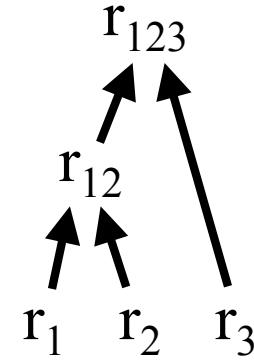
$B_{1,1}$ r_1	$B_{1,2}$ r_2, r_3
$B_{2,1}$ r_1, r_2	$B_{2,2}$ r_3



Block Queue



Current Block



$$r_1 \Rightarrow r_1$$

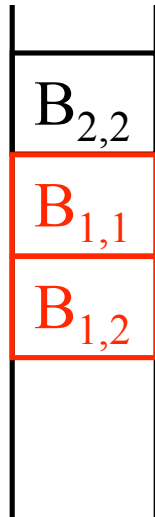
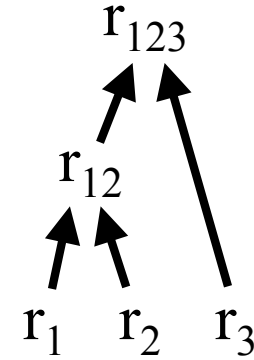
$$r_2 \Rightarrow r_2$$

$$r_3 \Rightarrow r_3$$

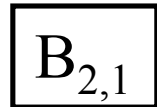
Maximal Records

Example

$B_{1,1}$ r_1	$B_{1,2}$ r_2, r_3
$B_{2,1}$ r_{12}	$B_{2,2}$ r_3



Block Queue



Current Block

$$r_1 \Rightarrow r_{12}$$

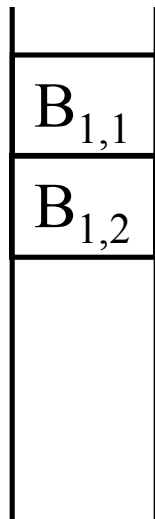
$$r_2 \Rightarrow r_{12}$$

$$r_3 \Rightarrow r_3$$

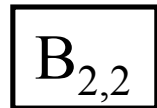
Maximal Records

Example

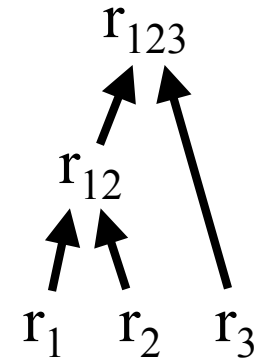
$B_{1,1}$ r_1	$B_{1,2}$ r_2, r_3
$B_{2,1}$ r_{12}	$B_{2,2}$ r_3



Block Queue



Current Block



$$r_1 \Rightarrow r_{12}$$

$$r_2 \Rightarrow r_{12}$$

$$r_3 \Rightarrow r_3$$

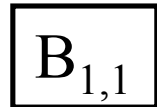
Maximal Records

Example

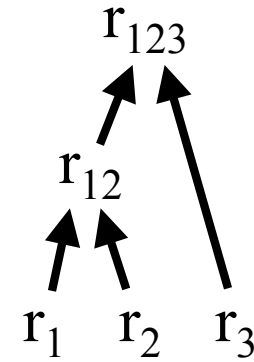
$B_{1,1}$ r_1	$B_{1,2}$ r_2, r_3
$B_{2,1}$ r_{12}	$B_{2,2}$ r_3



Block Queue



Current Block



$$r_1 \Rightarrow r_{12}$$

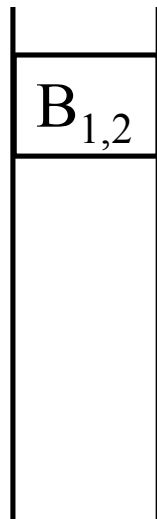
$$r_2 \Rightarrow r_{12}$$

$$r_3 \Rightarrow r_3$$

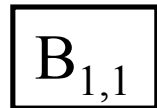
Maximal Records

Example

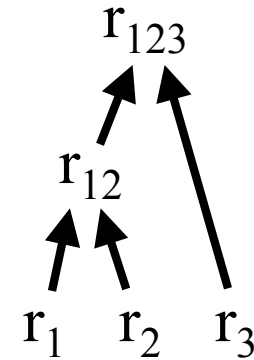
$B_{1,1}$ r_{12}	$B_{1,2}$ r_2, r_3
$B_{2,1}$ r_{12}	$B_{2,2}$ r_3



Block Queue



Current Block



$$r_1 \Rightarrow r_{12}$$

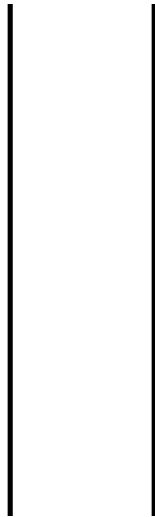
$$r_2 \Rightarrow r_{12}$$

$$r_3 \Rightarrow r_3$$

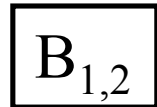
Maximal Records

Example

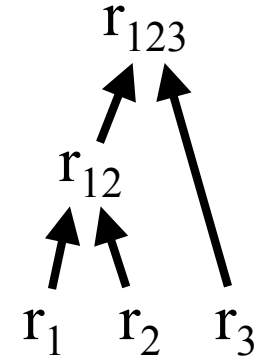
$B_{1,1}$ r_{12}	$B_{1,2}$ r_2, r_3
$B_{2,1}$ r_{12}	$B_{2,2}$ r_3



Block Queue



Current Block



$$r_1 \Rightarrow r_{12}$$

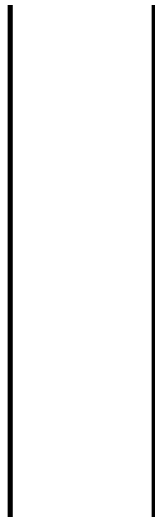
$$r_2 \Rightarrow r_{12}$$

$$r_3 \Rightarrow r_3$$

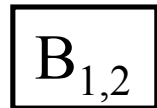
Maximal Records

Example

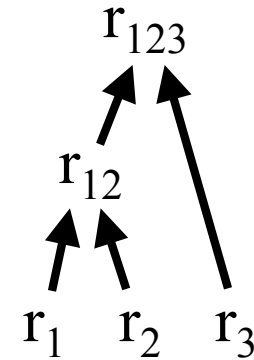
$B_{1,1}$ r_{12}	$B_{1,2}$ r_{12}, r_3
$B_{2,1}$ r_{12}	$B_{2,2}$ r_3



Block Queue



Current Block



$$r_1 \Rightarrow r_{12}$$

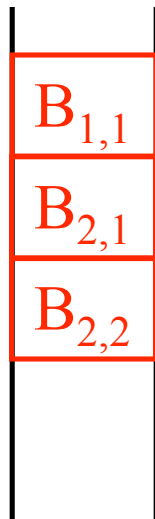
$$r_2 \Rightarrow r_{12}$$

$$r_3 \Rightarrow r_3$$

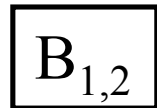
Maximal Records

Example

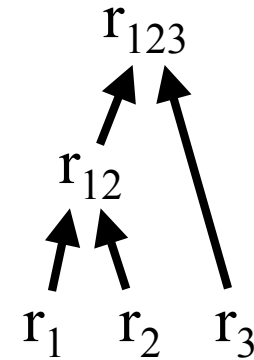
$B_{1,1}$ r_{12}	$B_{1,2}$ r_{123}
$B_{2,1}$ r_{12}	$B_{2,2}$ r_3



Block Queue



Current Block



$$r_1 \Rightarrow r_{123}$$

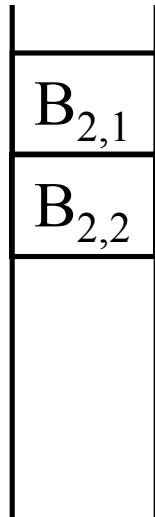
$$r_2 \Rightarrow r_{123}$$

$$r_3 \Rightarrow r_{123}$$

Maximal Records

Example

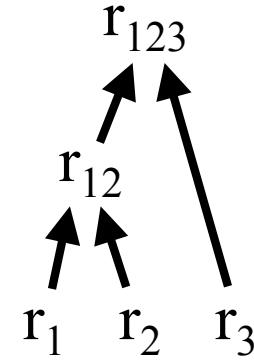
$B_{1,1}$ r_{123}	$B_{1,2}$ r_{123}
$B_{2,1}$ r_{12}	$B_{2,2}$ r_3



Block Queue



Current Block



$$r_1 \Rightarrow r_{123}$$

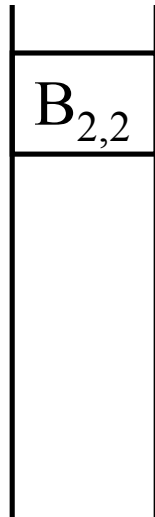
$$r_2 \Rightarrow r_{123}$$

$$r_3 \Rightarrow r_{123}$$

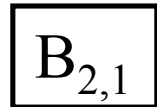
Maximal Records

Example

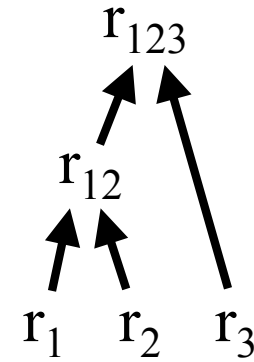
$B_{1,1}$ r_{123}	$B_{1,2}$ r_{123}
$B_{2,1}$ r_{123}	$B_{2,2}$ r_3



Block Queue



Current Block



$$r_1 \Rightarrow r_{123}$$

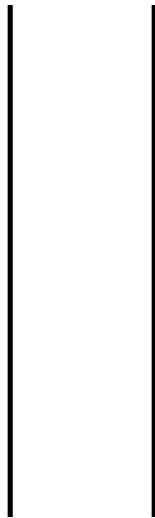
$$r_2 \Rightarrow r_{123}$$

$$r_3 \Rightarrow r_{123}$$

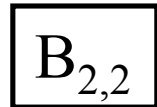
Maximal Records

Example

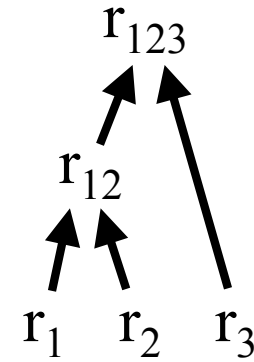
$B_{1,1}$ r_{123}	$B_{1,2}$ r_{123}
$B_{2,1}$ r_{123}	$B_{2,2}$ r_{123}



Block Queue



Current Block



$$r_1 \Rightarrow r_{123}$$

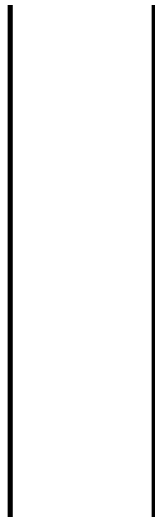
$$r_2 \Rightarrow r_{123}$$

$$r_3 \Rightarrow r_{123}$$

Maximal Records

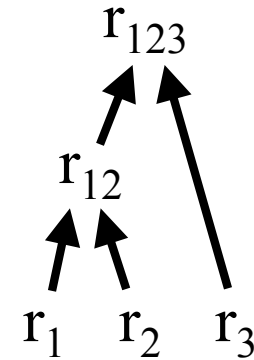
Example

$B_{1,1}$ r_{123}	$B_{1,2}$ r_{123}
$B_{2,1}$ r_{123}	$B_{2,2}$ r_{123}



Block Queue

Current Block



$r_1 \Rightarrow r_{123}$

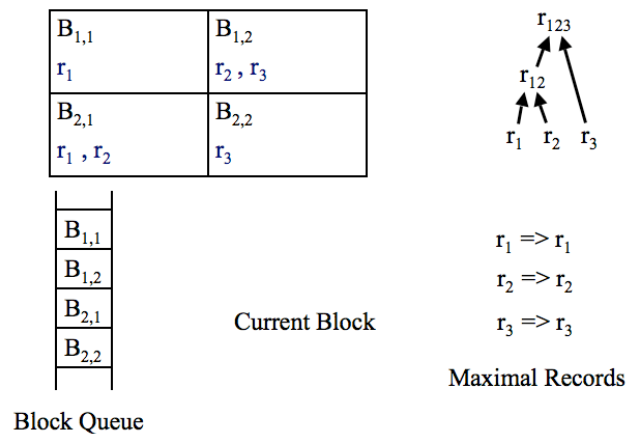
$r_2 \Rightarrow r_{123}$

$r_3 \Rightarrow r_{123}$

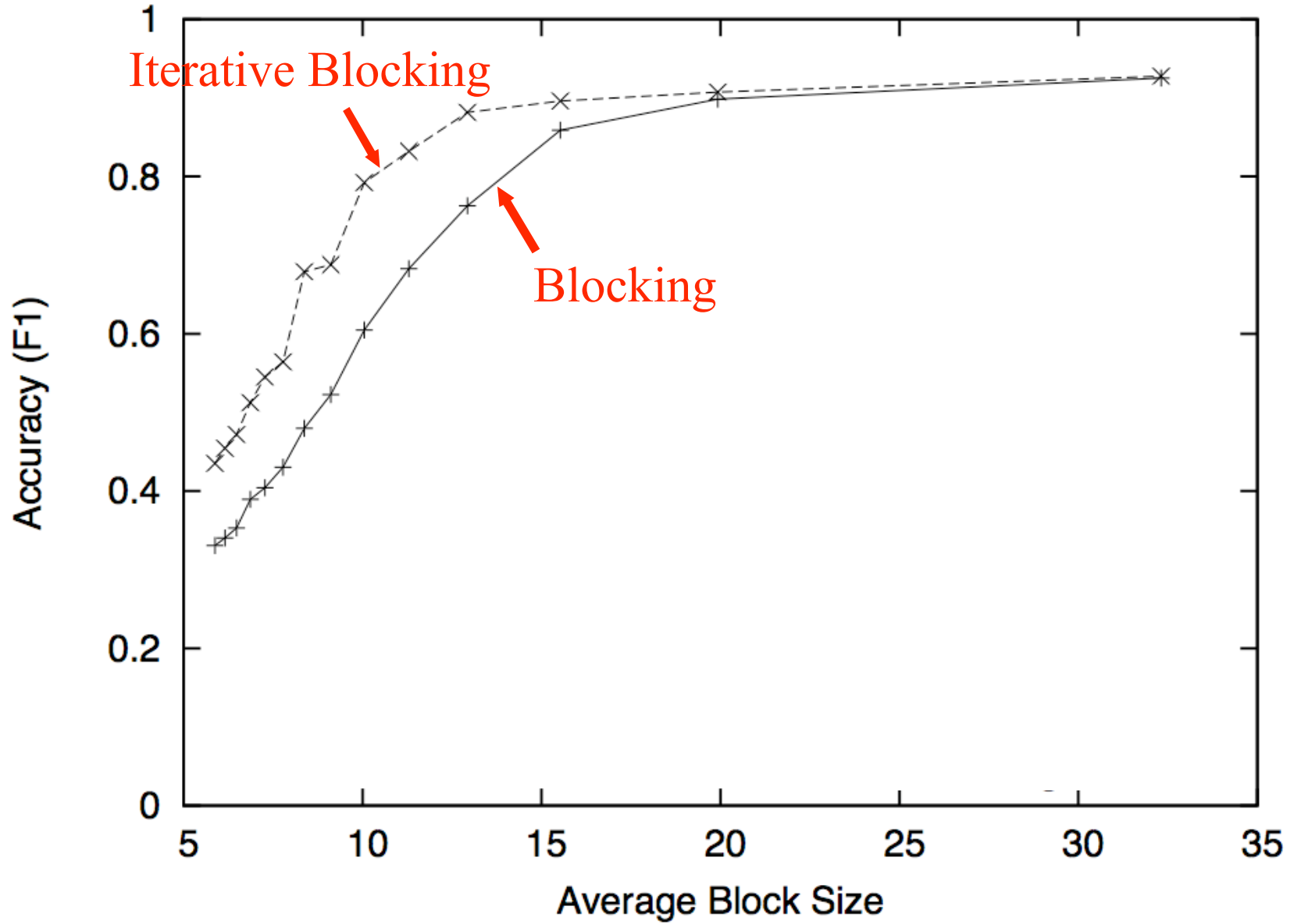
Maximal Records

Disk-based Algorithm (Duplo)

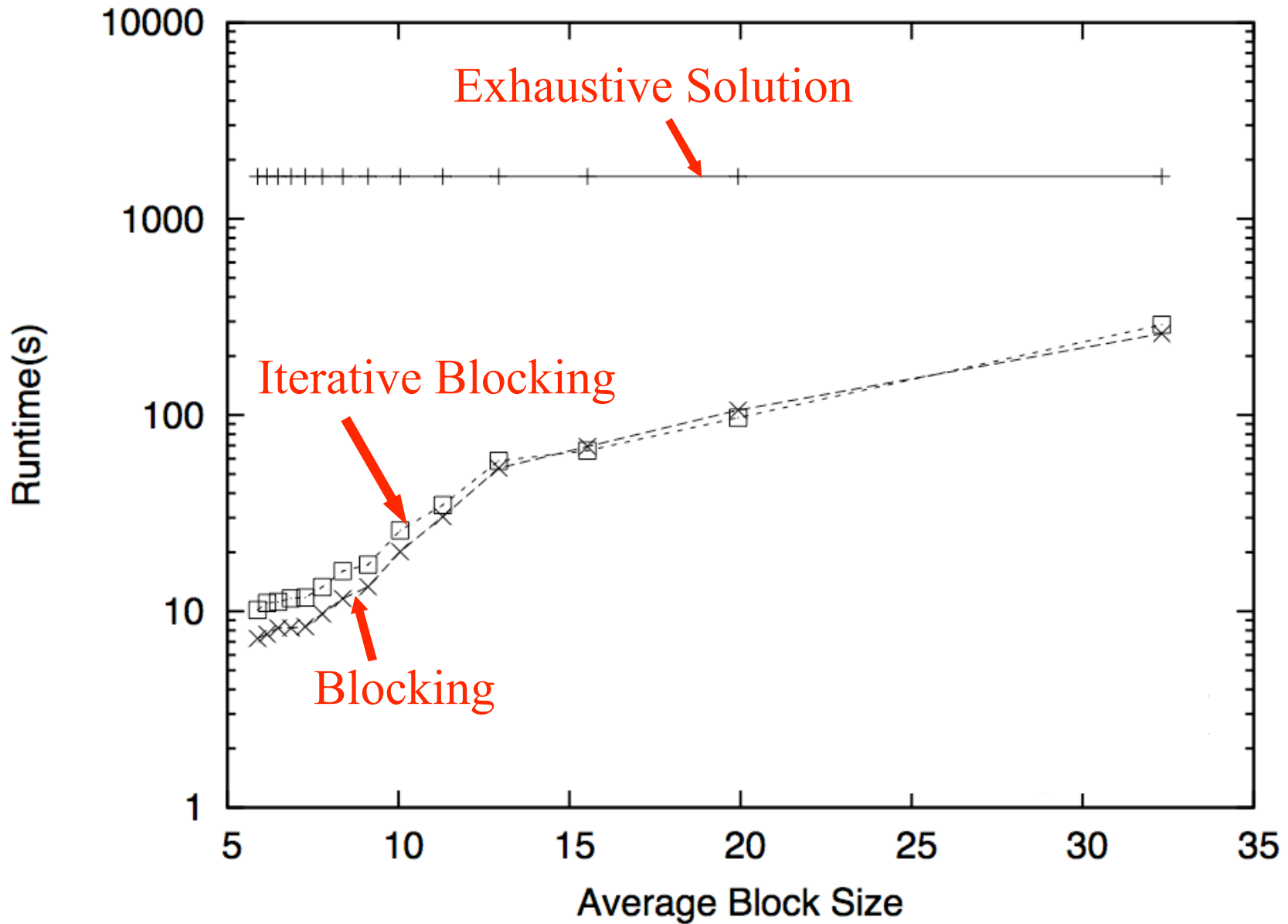
- Reads into memory and processes “N blocks at a time” using segments
- Maintains a queue of segments to process
- Updates segments by scanning a merge log on disk
 - Uses timestamps to avoid a full scan for each segment read



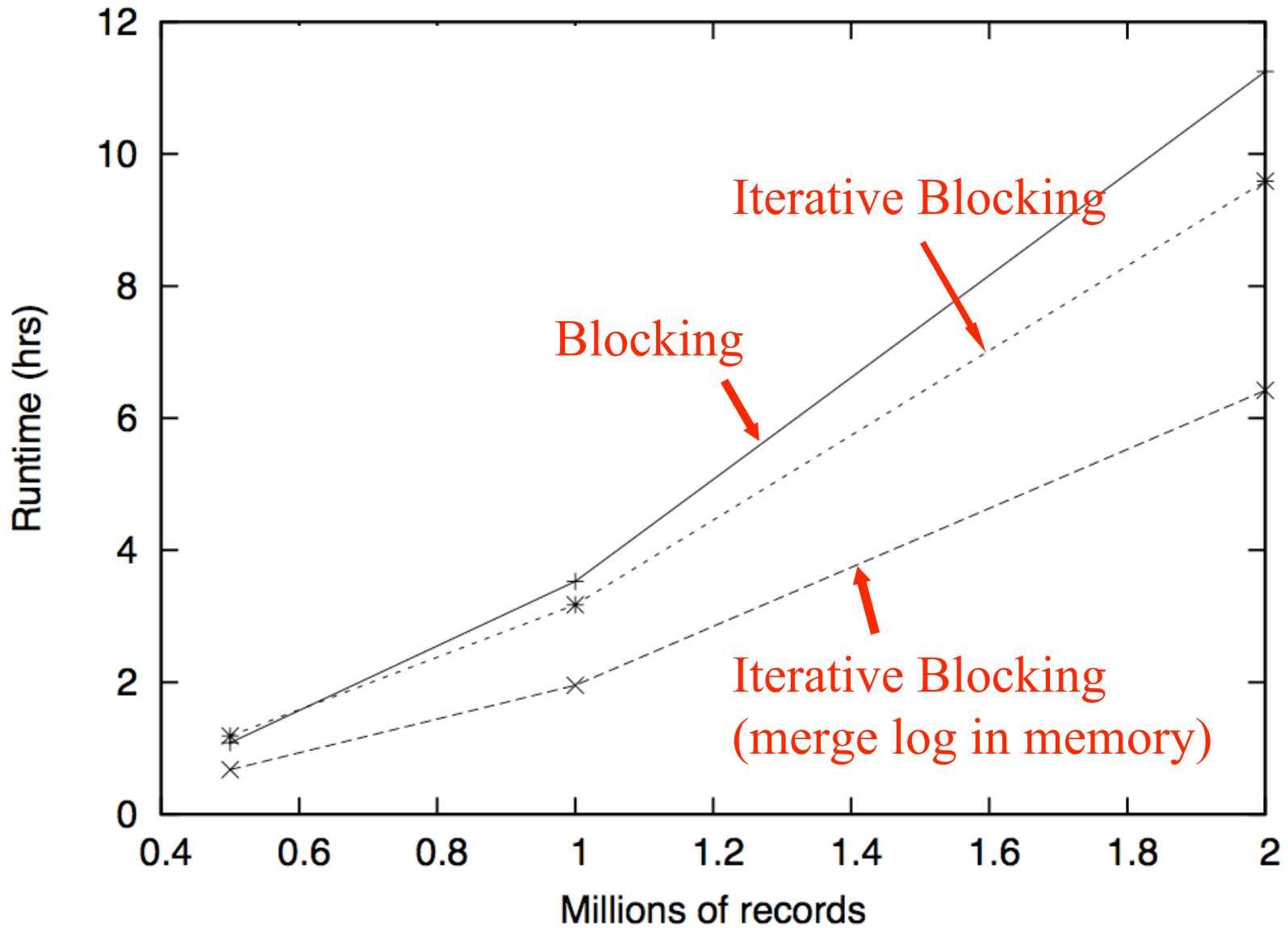
Accuracy



Performance



Scalability



Conclusion

- Proposed model & efficient algorithms (in-memory, disk) for iterative blocking
- Showed that iterative blocking can be more accurate and scalable than simple blocking

Questions?