Towards an Understanding of the Limits of Map-Reduce Computation

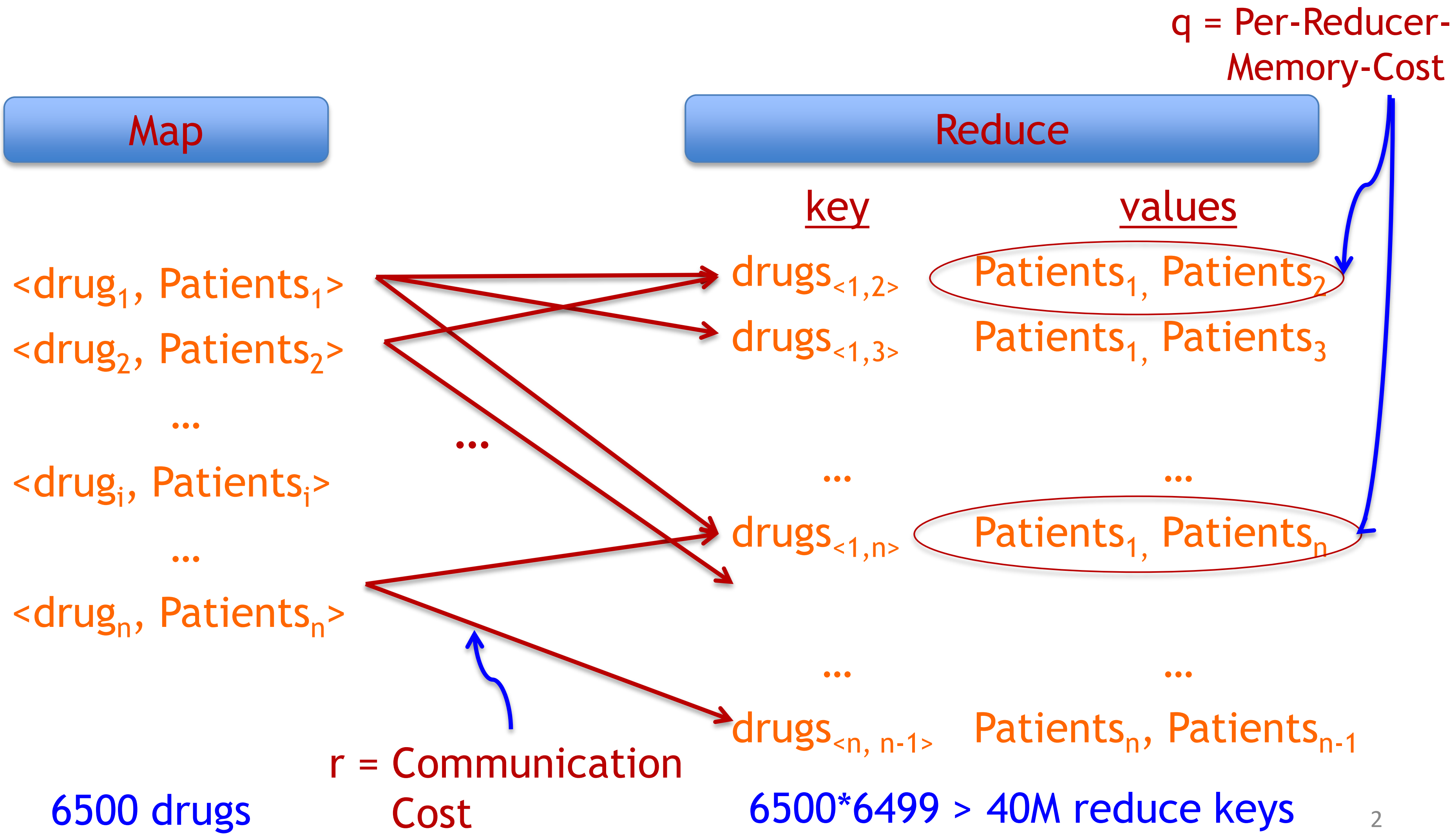
Foto Afrati — National Technical University of Athens

Anish Das Sarma — Google Research

Semih Salihoglu — Stanford University

Jeff Ullman — Stanford University

Tradeoff Between *Per-Reducer-Memory* and *Communication Cost*



q = Per-Reducer-Memory-Cost

Map

Reduce

key

values

<drug$_1$, Patients$_1$>

<drug$_2$, Patients$_2$>

…

<drug$_i$, Patients$_i$>

…

<drug$_n$, Patients$_n$>

…

drugs$_{<1,2>}$ Patients$_1$, Patients$_2$

drugs$_{<1,3>}$ Patients$_1$, Patients$_3$

… …

drugs$_{<1,n>}$ Patients$_1$, Patients$_n$

… …

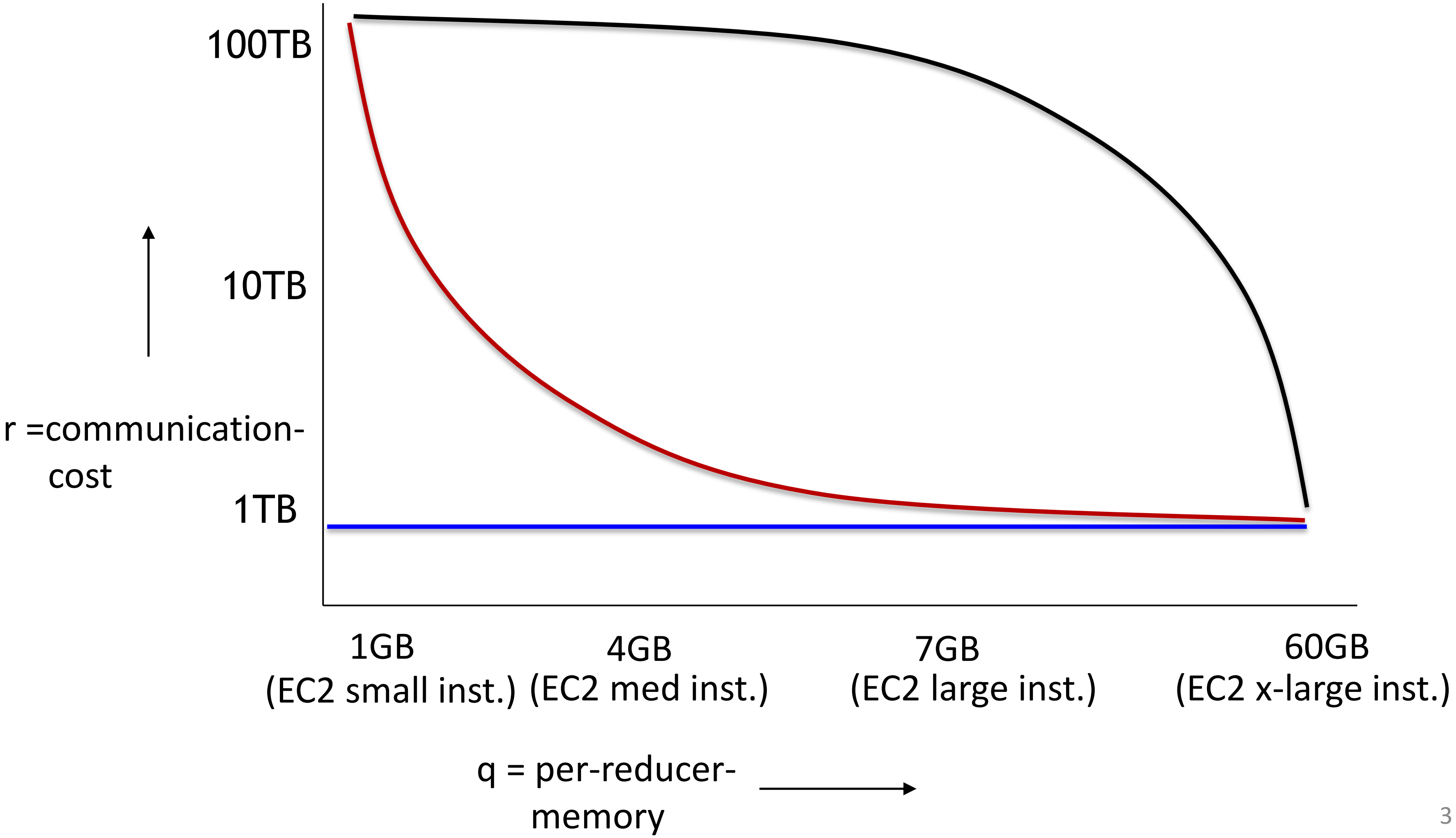
drugs$_{<n, n-1>}$ Patients$_n$, Patients$_{n-1}$

r = Communication Cost

6500 drugs

6500*6499 > 40M reduce keys

Possible Per-Reducer-Memory/ Communication Cost Tradeoffs



Example (1)

- Similarity Join
  - Input R(A, B), Domain(B) = [1, 10]
  - Compute <t, u> s.t |t[B]-u[B]| · 1

Input

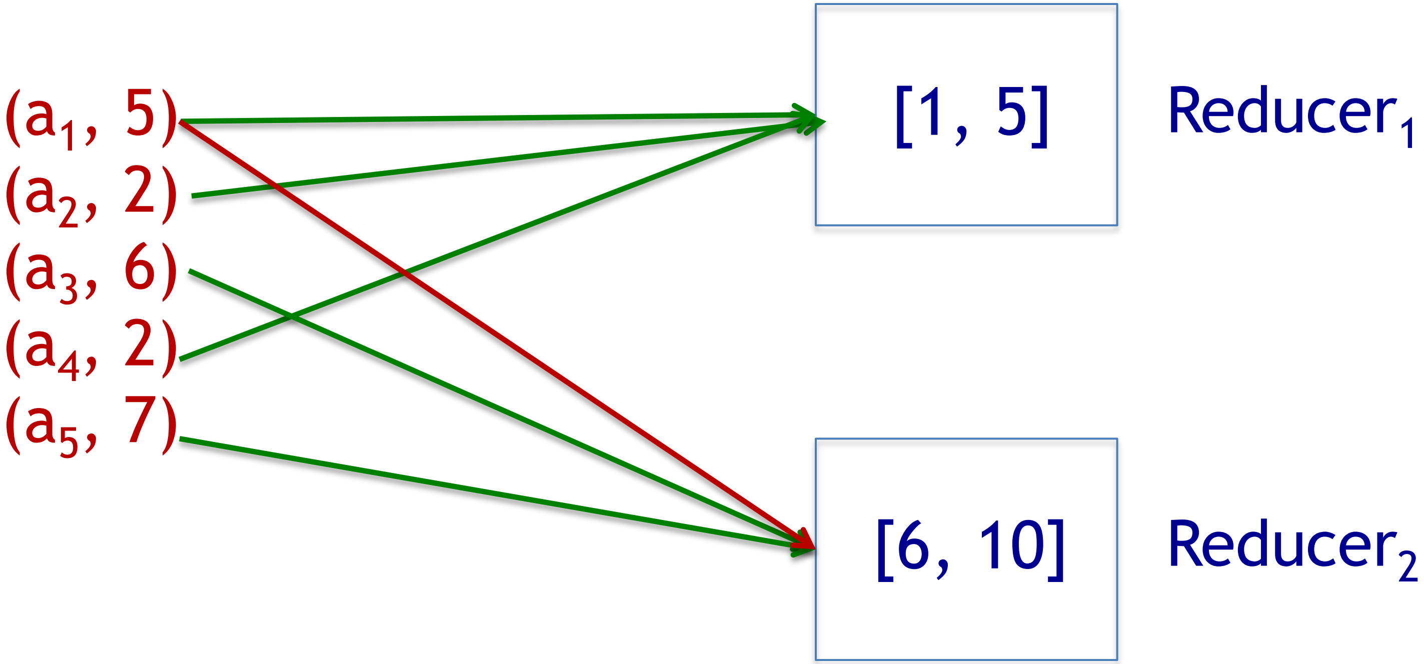
A	B
a_1	5
a_2	2
a_3	6
a_4	2
a_5	7

Output

<$(a_1, 5)$, $(a_3, 6)$>
<$(a_2, 2)$, $(a_4, 2)$>
<$(a_3, 6)$, $(a_5, 7)$>

Example (2)

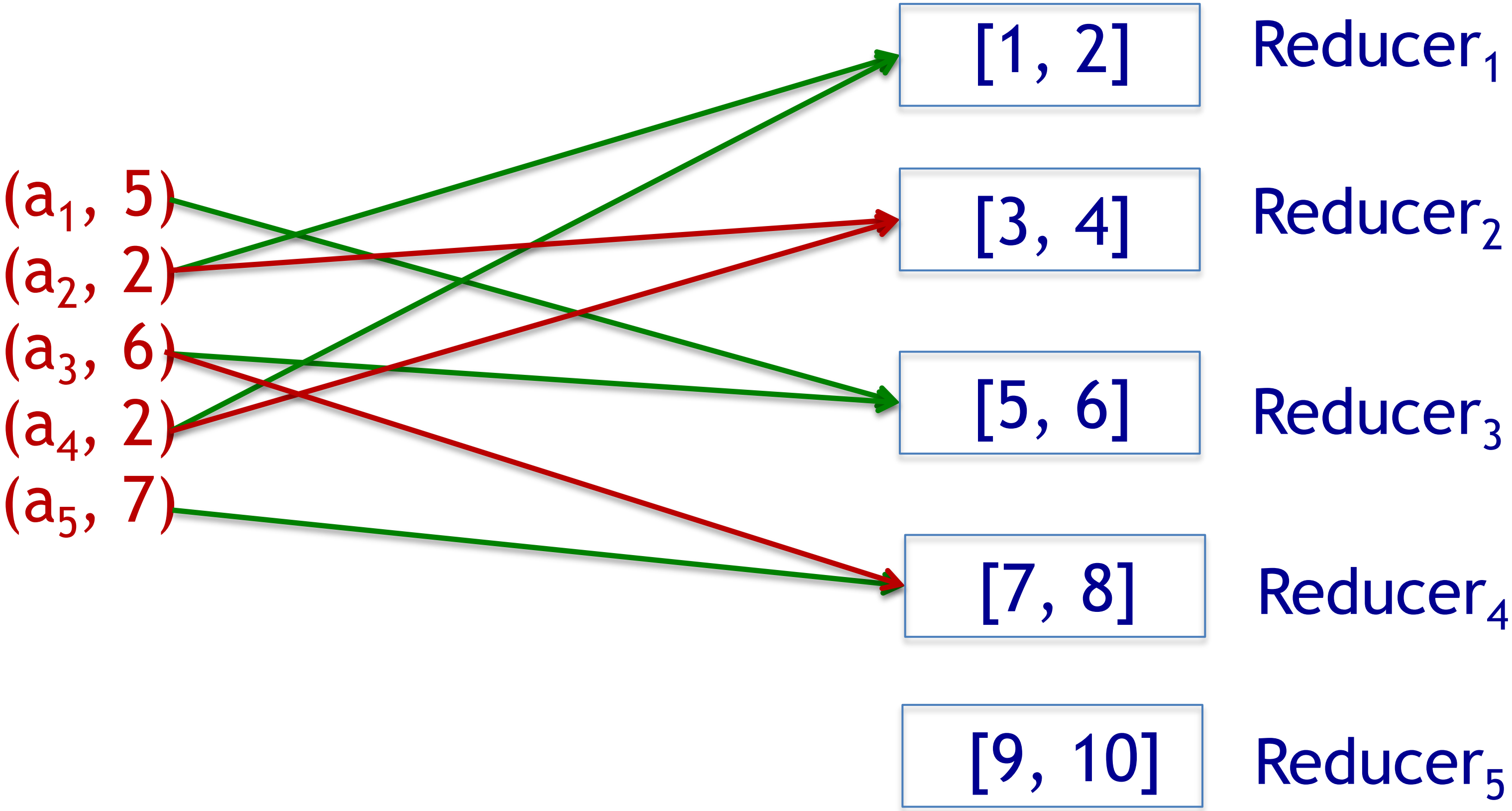
- Hashing Algorithm [ADMPU ICDE ’12]
- Split Domain(B) into k ranges of values => (k reducers)
- k = 2



- Replicate tuples on the boundary (if t.B = 5)
- Per-Reducer-Memory Cost = 3, Communication Cost = 6

Example (3)

- k = 5 => Replicate if t.B = 2, 4, 6 or 8



- Per-Reducer-Memory Cost = 2, Communication Cost = 8

Same Tradeoff in Other Algorithms

- Finding subgraphs ([SV] WWW ’11, [AFU] Tech Report ’12)
- Computing Minimum Spanning Tree (KSV SODA ’10)
- Other similarity joins:
    - Set similarity joins ([VCL] SIGMOD ’10)
    - Hamming Distance (ADMPU ICDE ’12 and later in the talk)

Our Goals

- General framework for studying memory/communication tradeoff, applicable to a variety of problems
- Question 1: What is the *minimum* communication for *any* MR algorithm, if each reducer uses ▪ *q* memory?
- Question 2: Are there algorithms that achieve this lower bound?

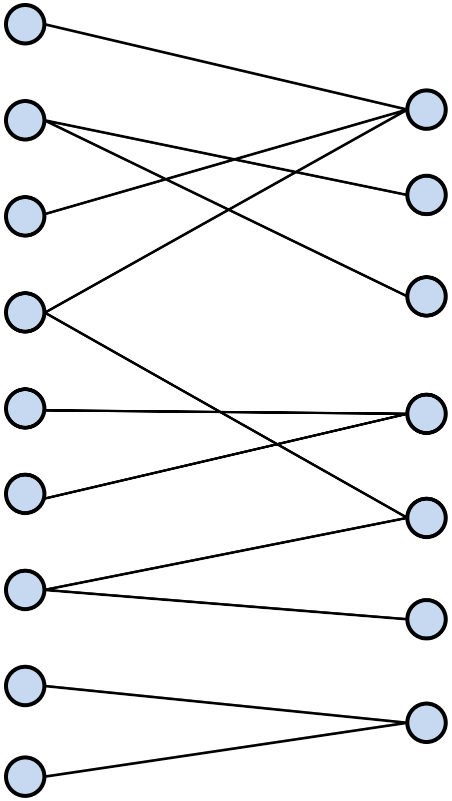
Remainder of Talk

- Input-Output Model
- Mapping Schemas & Replication Rate
- Hamming Distance 1
- Other Results

Input-Output Model

Input Data Elements

I: $\{i_1, i_2, \ldots, i_n\}$



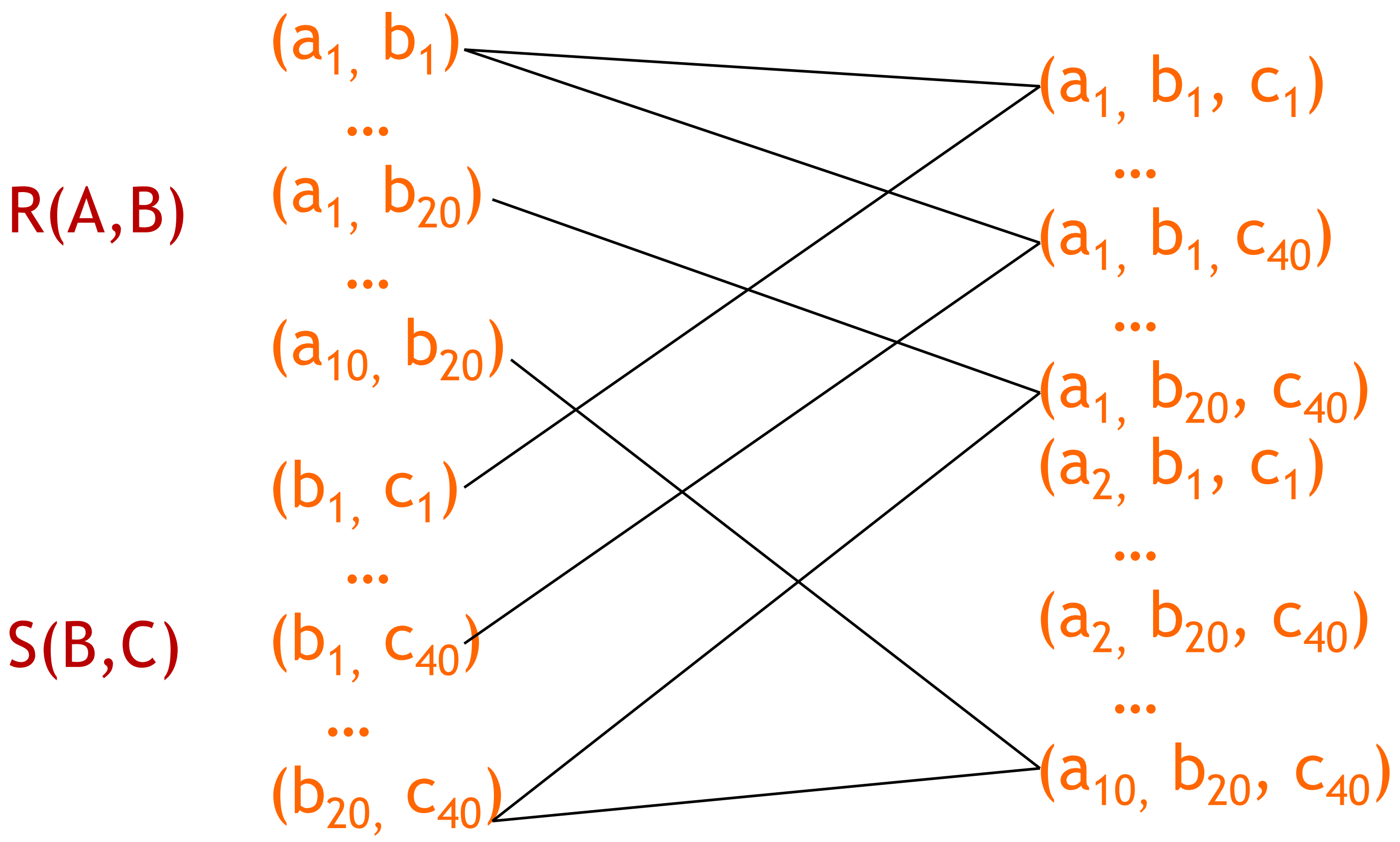
Output Elements

O: $\{o_1, o_2, \ldots, o_m\}$

Dependency = Provenance

Example 1: R(A, B) $\bowtie$ S(B, C)

- |Domain(A)| = 10, |Domain(B)| = 20, |Domain(C)| = 40



R(A,B)

(a_1, b_1)

…

(a_1, b_{20})

…

(a_{10}, b_{20})

S(B,C)

(b_1, c_1)

…

(b_1, c_{40})

…

(b_{20}, c_{40})

(a_1, b_1, c_1)

…

(a_1, b_1, c_{40})

…

(a_1, b_{20}, c_{40})

(a_2, b_1, c_1)

…

(a_2, b_{20}, c_{40})

…

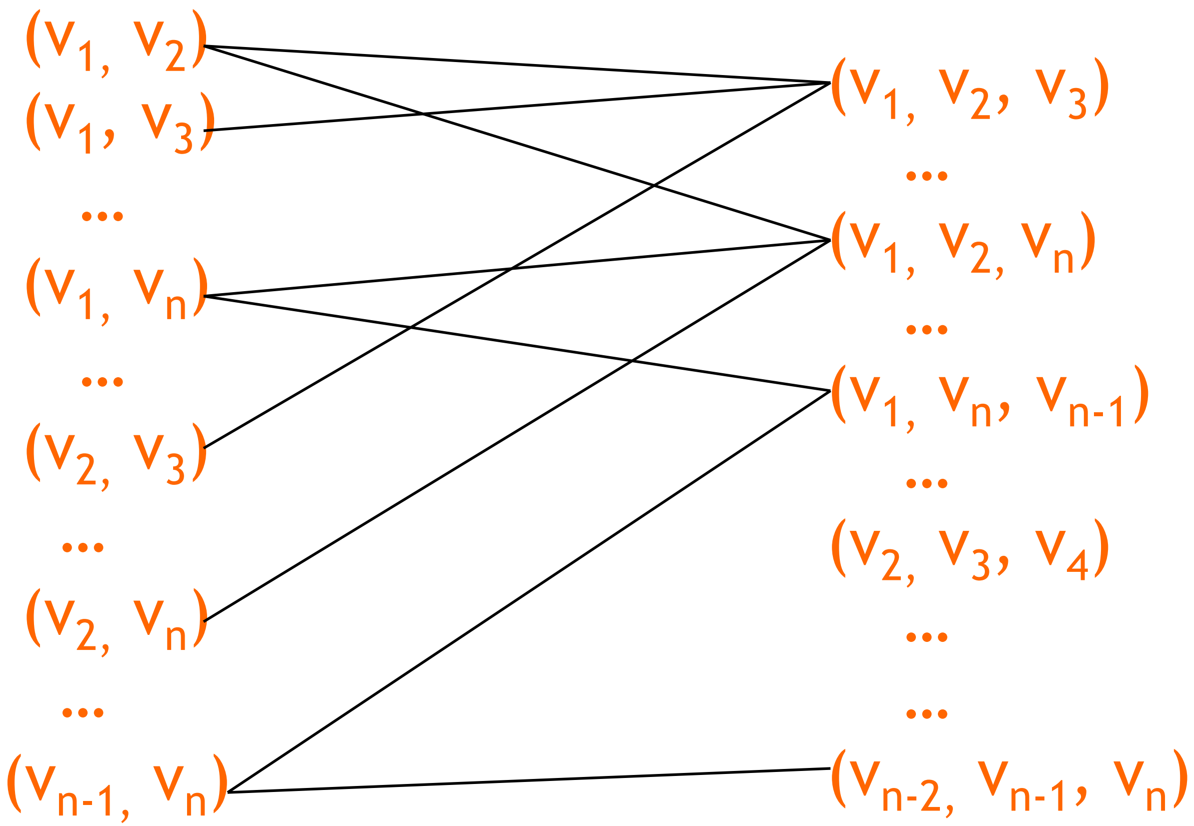
(a_{10}, b_{20}, c_{40})

10*20 + 20*40 =
1000 input elements

10*20*40 =
8000 output elements

Example 2: Finding Triangles

- Graphs G(V, E) of n vertices $\{v_1, \ldots, v_n\}$



(v_1, v_2)

(v_1, v_3)

…

(v_1, v_n)

…

(v_2, v_3)

…

(v_2, v_n)

…

(v_{n-1}, v_n)

(v_1, v_2, v_3)

…

(v_1, v_2, v_n)

…

(v_1, v_n, v_{n-1})

…

(v_2, v_3, v_4)

…

…

(v_{n-2}, v_{n-1}, v_n)

n-choose-2
input data elements

n-choose-3
output elements

Mapping Schema & Replication Rate

- *p* reducer: $\{R_1, R_2, \ldots, R_p\}$
- *q* max # inputs sent to any reducer R_i
- Def (Mapping Schema): *M* : *I* → $\{R_1, R_2, \ldots, R_p\}$ s.t
  - R_i receives at most q_i · q inputs
  - Every output is *covered* by some reducer:
- Def (Replication Rate):
  - $r = \sum_{i=1}^{p} q_i \Big/ |I|$
- q captures memory, r captures communication cost

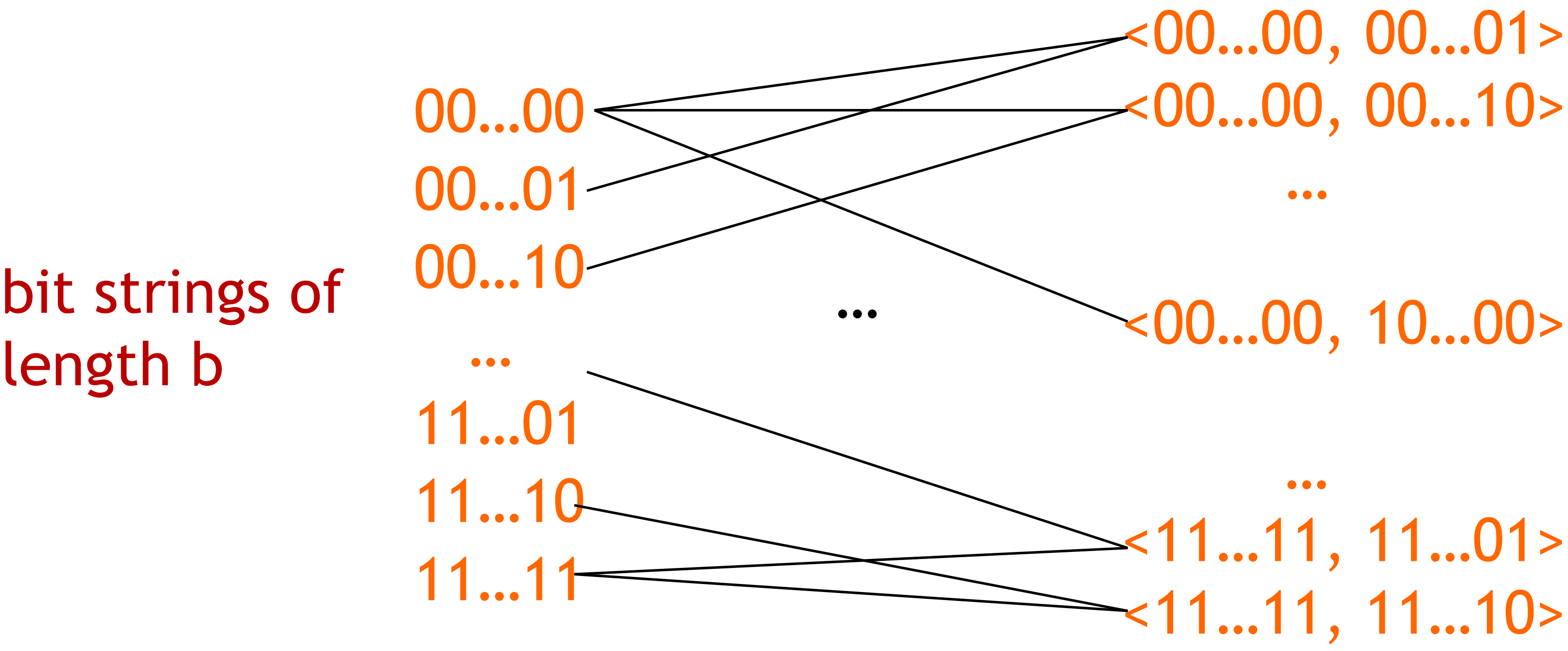
Our Questions Again

- Question 1: What is the minimum replication rate of any mapping schema as a function of q (maximum # inputs sent to any reducer)?
- Question 2: Are there mapping schemas that match this lower bound?

Hamming Distance = 1

each input *contributes* to b outputs

each output depends on 2 inputs



bit strings of length b

00…00

00…01

00…10

…

11…01

11…10

11…11

…

<00…00, 00…01>

<00…00, 00…10>

…

<00…00, 10…00>

…

<11…11, 11…01>

<11…11, 11…10>

$|I| = 2^b$

$|O| = b2^b/2$

Lower Bound on Replication Rate (HD=1)

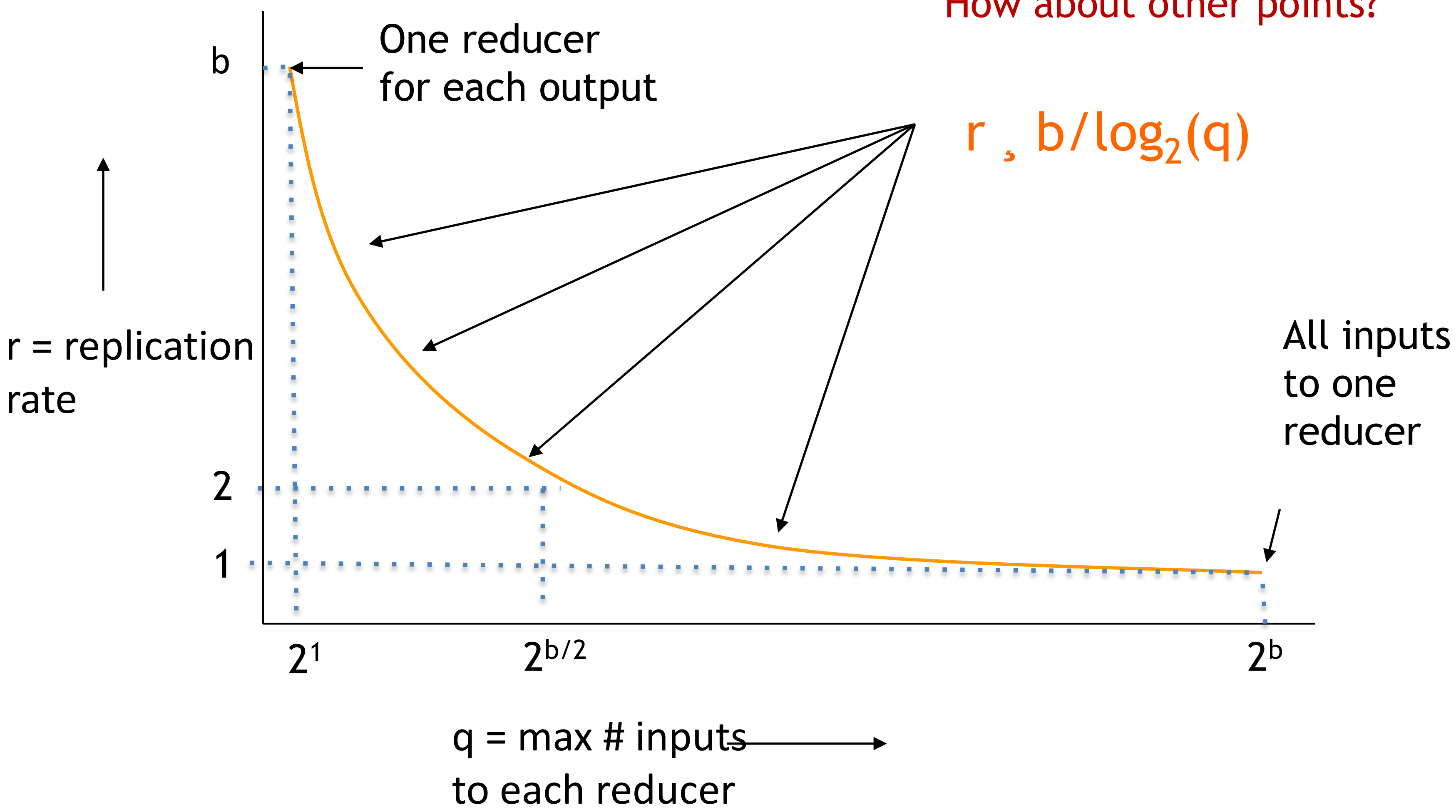
- Key is upper bound $g(q)$: max outputs a reducer can cover with · q inputs
- Claim: $g(q) = \frac{q}{2}\log_2(q)$ (proof by induction on b)
- All outputs must be covered:

$$\mathop{\text{å}}\limits_{i=1}^{p} g(q_i) \text{ ³ } |O| \longrightarrow \mathop{\text{å}}\limits_{i=1}^{p} \frac{q_i}{2}\log_2 q_i \text{ ³ } \frac{b}{2}2^b \longrightarrow \mathop{\text{å}}\limits_{i=1}^{p} \frac{q_i}{2}\log_2 q \text{ ³ } \frac{b}{2}2^b$$

- Recall: $\text{r} = \mathop{\text{å}}\limits_{i=1}^{p} q_i \Big/ |I| \longrightarrow \text{r} = \mathop{\text{å}}\limits_{i=1}^{p} q_i \Big/ 2^b$

r ¸ b/$\log_2$(q)

Memory/Communication Cost Tradeoff (HD=1)



How about other points?

One reducer for each output

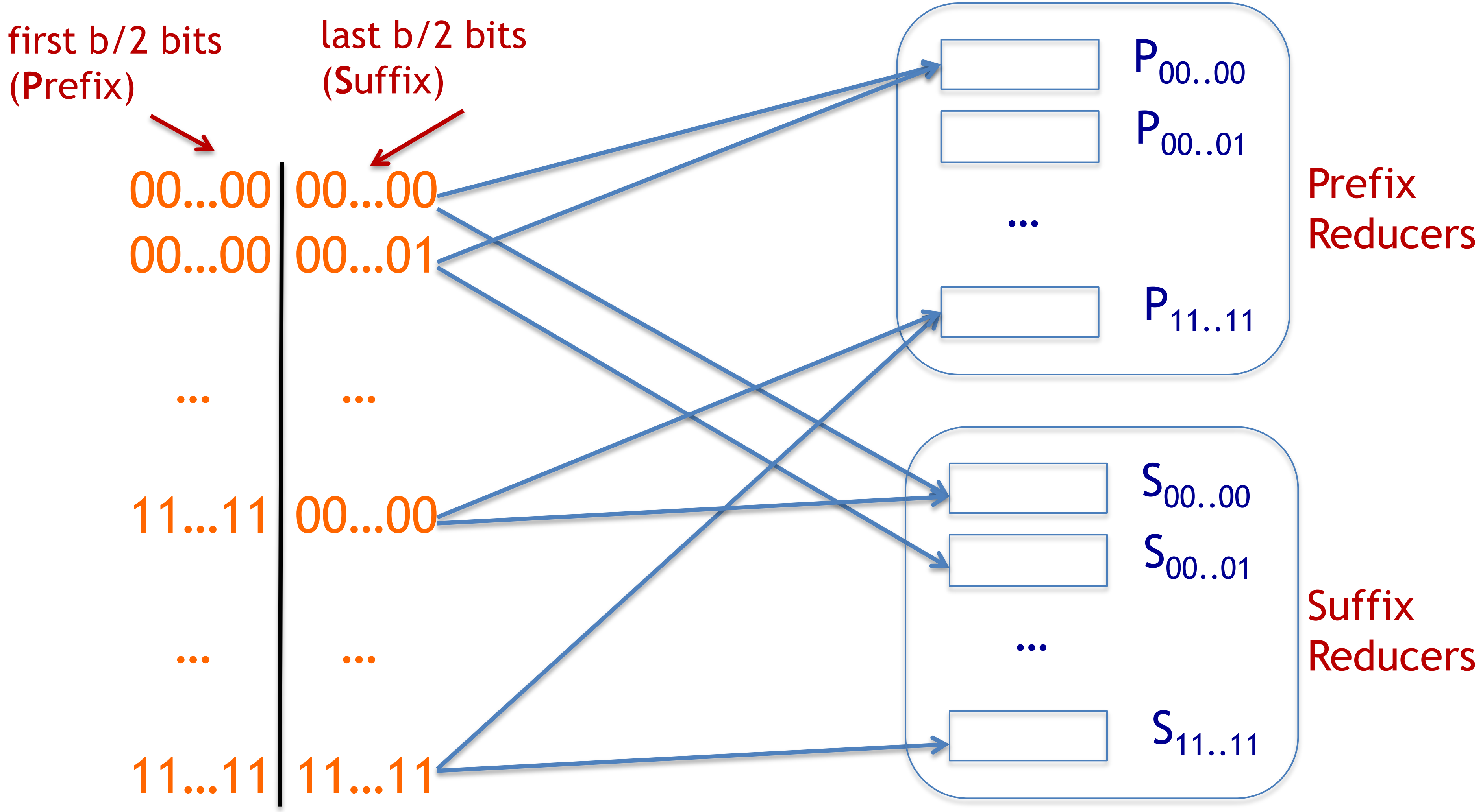
r ¸ $b/\log_2(q)$

All inputs to one reducer

r = replication rate

q = max # inputs to each reducer

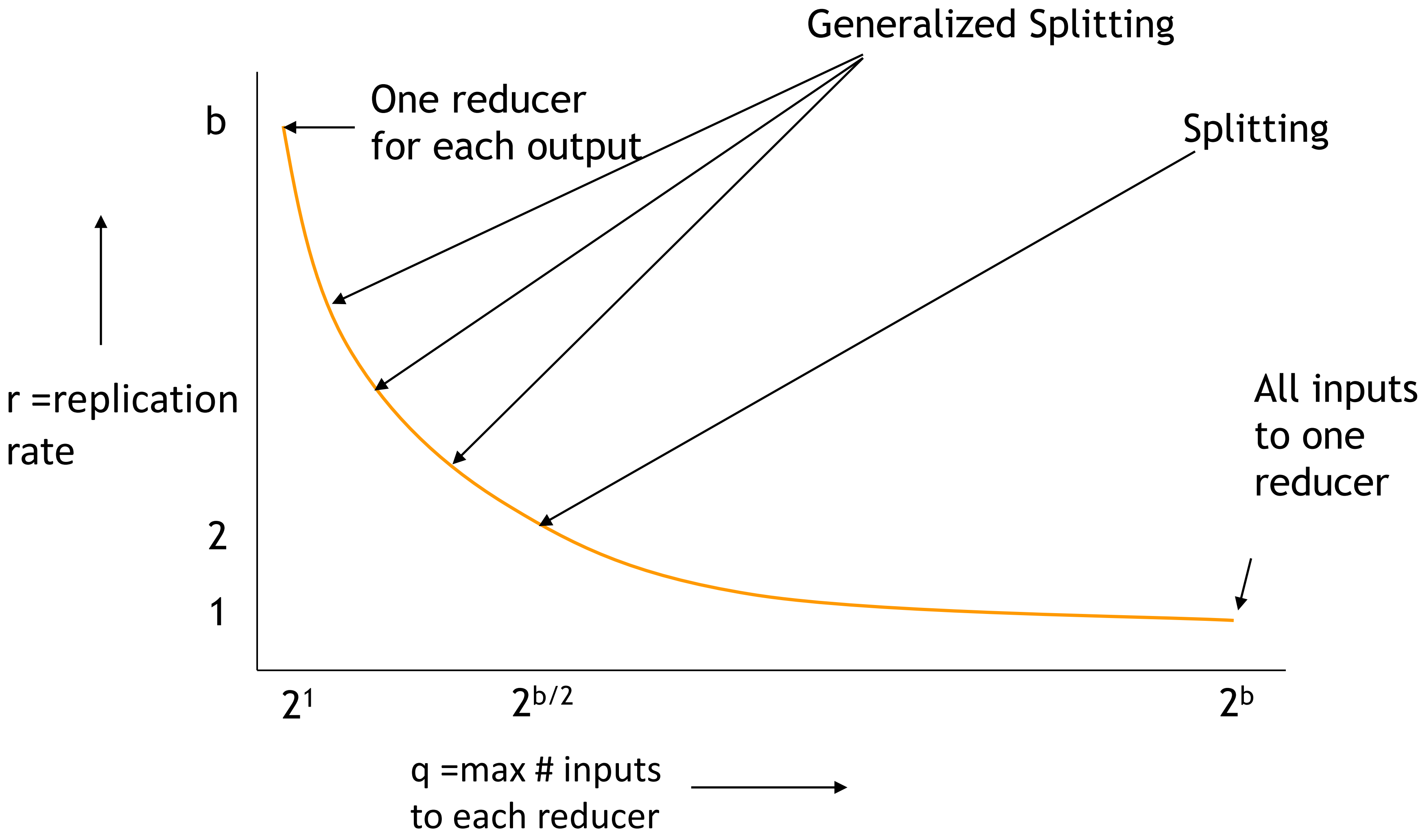
Splitting Algorithm for HD 1 ($q=2^{b/2}$)



$r=2$, $q=2^{b/2}$

$2^{b/2} + 2^{b/2}$ Reducers

Where we stand for HD = 1



General Method for Using Our Framework

1. Represent problem *P* in terms of *I*, *O*, and dependencies
2. Lower bound for r as function of q:
   - i. Upper bound on $g(q)$: max outputs covered by q inputs
   - ii. All outputs must be covered: $\sum_{1}^{p} g(q_i) \geq |O|$
   - iii. Manipulate (ii) to get r = $\sum_{1}^{p} q_i / |I|$ as a function of q
3. Demonstrate algorithms/mapping schemas that match the lower bound

Other Results

- Finding Triangles in G(V, E) with n vertices:
  - Lower bound: r ¸ $\frac{n}{\sqrt{2q}}$
  - Algorithms: $O(\frac{n}{\sqrt{2q}})$
- Multiway Self Joins:
  - R(A_{11},…,A_{1k}) ⋈ R(A_{21},…, A_{2k}) ⋈ … ⋈ R(A_{t1},…, A_{tk})
  - *k* # columns, *n* = |A_i|, join *t* times on *i* columns
  - Lower bound & Algorithms: $O(q^{1-t(k-i)/k} n^{t(k-i)-k})$
- Hamming distance · d
  - Algorithms: r · d + 1

Related Work

- Efficient Parallel Set Similarity Joins Using MapReduce (Vernica, Carey, Li in SIGMOD '10)
- Processing Theta Joins Using MapReduce (Okcan, Riedewald in SIGMOD '11)
- Fuzzy Joins Using MapReduce (Afrati, Das Sarma, Menestrina, Parameswaran, Ullman in ICDE '12)
- Optimizing Joins in a MapReduce Environment (Afrati, Ullman in EDBT '10)
- Counting Triangles and the Curse of the Last Reducer (Suri, Vassilvitskii WWW '11)
- Enumerating Subgraph Instances Using MapReduce (Afrati, Fotakis, Ullman as Techreport 2011)
- A Model of Computation for MapReduce (Karloff, Suri, Vassilvitskii in SODA '10)

Future Work

- Derive lower bounds on replication rate and match this lower bound with algorithms for many different problems.
- Relate structure of input-output dependency graph to replication rate.
    - How does min-cut size relate to replication rate?
    - How does expansion rate relate to replication rate?