

- Background
- Model
- Hamming Distance 1
- Triangle Finding
- Matrix Multiplication

# Agenda

1

## The Problem

- Tradeoff between parallelism and communication cost in a map-reduce computation.
- The finer we partition the work of the reducers so that more parallelism can be extracted, the greater will be the total communication between mappers and reducers.
- Limited bandwidth
- Limited resources(memory, processing units...)

# Background

Why important

- Explore the bounds on the cost of map-reduce computation.
- Optimize the algorithms for problem.

# Background

3

## Previous Work

- First work that addresses the tradeoff between reducer size and communication cost in one round Map-Reduce computations.
- Theta-join implementation by Map-Reduce: only one special case.
- Limit the input size of any reducer: limits consideration to algorithms that we might think of as truly parallel.
- ...

# Background

- A model of problems that can be solved in a single round of map-reduce computation.

## Two Parameters

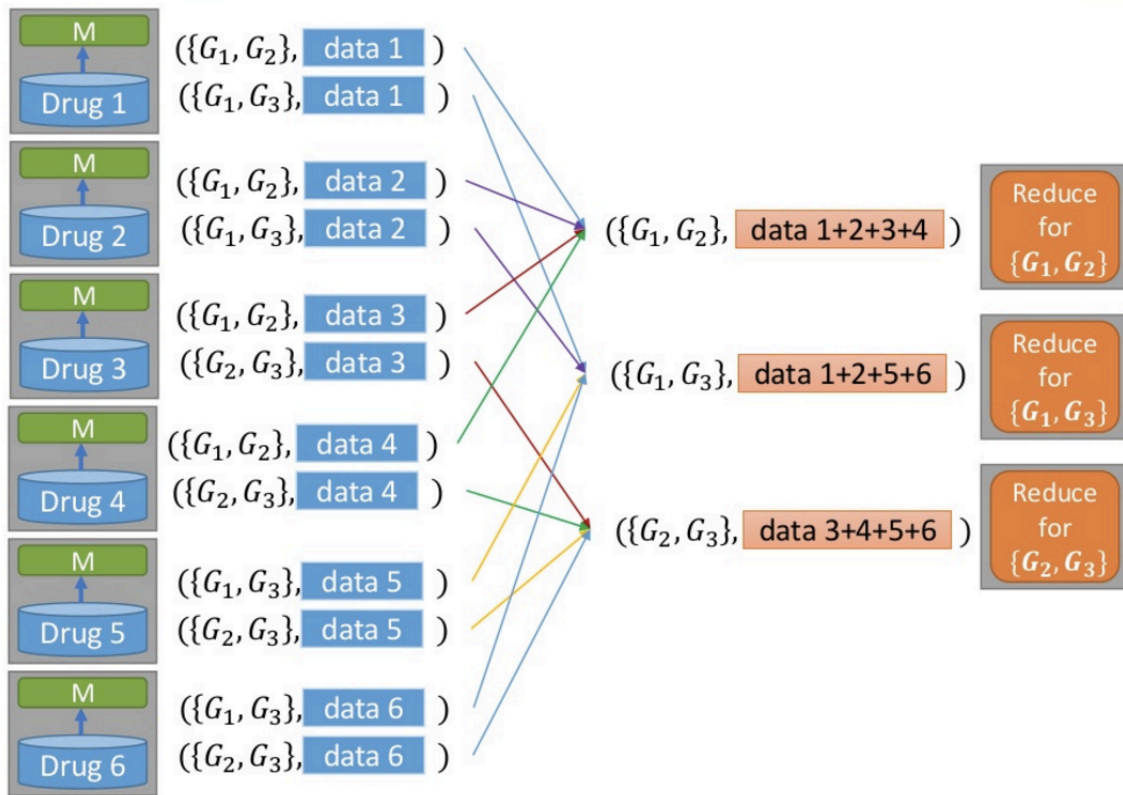
- Replication rate  $r$ : average number of key-value pairs to which each input is mapped by the mappers.
- Reducer size  $p$ : the maximum number of inputs that one reducer can receive.

# Model

$$r = 2$$

$$p = 4$$

# Model



## Tradeoff

- Determine the best algorithm for a problem where:

$$r = f(q)$$

- Cost of solving the problem:

$$af(q) + bq(+cq^2)$$

- Replication rate:

$$r = \sum_{i=1}^p \frac{q_i}{I}$$

# Model

## Mapping Schemas

- No reducer is assigned more than  $q$  inputs.
- For every output, there is (at least) one reducer that is assigned all of the inputs for that output. We say such a reducer covers the output. This reducer need not be unique, and it is permitted that these same inputs are assigned also to other reducers.

# Model



1. **Deriving  $g(q)$ :** First, find an upper bound,  $g(q)$ , on the number of outputs a reducer can cover if  $q$  is the number of inputs it is given.
2. **Number of Inputs and Outputs:** Count the total numbers of inputs  $|I|$  and outputs  $|O|$ .
3. **The Inequality:** Assume there are  $p$  reducers, each receiving  $q_i \leq q$  inputs and covering  $g(q_i)$  outputs. Together they cover all the outputs. That is:

$$\sum_{i=1}^p g(q_i) \geq |O| \quad (1)$$

4. **Replication Rate:** Manipulate the inequality from Equation [1] to get a lower bound on the replication rate, which is  $\sum_{i=1}^p q_i / |I|$ .

Note that the last step above may require clever manipulation to factor out the replication rate. We have noticed that the following “trick” is effective in Step (4) for all problems considered in this paper. First, arrange to isolate a single factor  $q_i$  from  $g(q_i)$ ; that is:

$$\sum_{i=1}^p g(q_i) \geq |O| \Rightarrow \sum_{i=1}^p q_i \frac{g(q_i)}{q_i} \geq |O| \quad (2)$$

Assuming  $\frac{g(q_i)}{q_i}$  is monotonically increasing in  $q_i$ , we can use the fact that  $\forall q_i : q_i \leq q$  to obtain from Equation [2]

$$\sum_{i=1}^p q_i \frac{g(q)}{q} \geq |O| \quad (3)$$

Now, divide both sides of Equation [3] by the input size, to get a formula with the replication rate on the left:

$$r = \frac{\sum_{i=1}^p q_i}{|I|} \geq \frac{q|O|}{g(q)|I|} \quad (4)$$

# Model

## Steps:

- Suppose the instance of the problem has  $|I|$  inputs and  $|O|$  outputs.
- We find an upper bound,  $g(q)$ , on the number of outputs any  $q$  inputs can generate.
- If  $g(q)/q$  is monotonically increasing in  $q$  then we can compute the replication rate using our recipe.
- Suppose the maximum number of inputs any reducer can take is  $q$ . Then the replication rate is  $r \geq \frac{q|O|}{g(q)|I|}$ .

# Model

Q1: Is this assumption reasonable?

Q2: Can be applied to most problems or only several specific problem?

Problem	$ I $	$ O $	$g(q)$	Lower bound on $r$
Hamming-Distance-1, $b$ -bit strings	$2^b$	$\frac{b2^b}{2}$	$\frac{q \log_2 q}{2}$ (Section 3.1)	$\frac{b}{\log_2 q}$ (Section 3.2)
Triangle-Finding, $n$ nodes	$\frac{n^2}{2}$	$\frac{n^3}{6}$	$\frac{\sqrt{2}}{3} q^{\frac{3}{2}}$ (Section 4.1)	$\frac{n}{\sqrt{2q}}$ (Section 4.1)
Sample Graphs (size $s$ nodes) in Alon Class in graph of $m$ edges, $n$ nodes	$\binom{n}{2}$ or $m$	$n^s$	$q^{s/2}$ (Section 5.2)	$(\frac{n}{\sqrt{q}})^{s-2}$ or $(\sqrt{\frac{m}{q}})^{s-2}$ (Sections 5.2 and 5.3)
2-Paths in $n$ -node graph	$\binom{n}{2}$	$\frac{n^3}{2}$	$\binom{q}{2}$ (Section 5.4.1)	$\frac{2n}{q}$ (Section 5.4.1)
Multiway Join: $N$ bin. rels, $m$ vars., Dom. $n$ , parameter $\rho$ from [7]	$N \binom{n}{2}$	$\binom{n}{m}$	$q^\rho$ ([7])	$\frac{n^{m-2}}{q^{\rho-1}}$ (Section 5.5.1)
$n \times n$ Matrix Multiplication	$2n^2$	$n^2$	$\frac{q^2}{4n^2}$ (Section 6.1)	$\frac{2n^2}{q}$ (Section 6.1)

Table 1: Lower bound on replication rate  $r$  for various problems in terms of number of inputs  $|I|$ , number of outputs  $|O|$ , and maximum number of inputs per reducer  $q$ .

# Model

Problem	Upper bound on $r$
Hamming-Distance-1 $b$ -bit strings	$\frac{b}{\log_2 q}$ (Section 3.3)
Triangle-Finding, $n$ nodes	$\mathcal{O}\left(\frac{n}{\sqrt{2q}}\right)$ (Section 4.2 and [2, 24])
Sample Graphs (size $s$ nodes) in Alon Class in graph of $m$ edges, $n$ nodes	$\mathcal{O}\left(\left(\sqrt{\frac{m}{q}}\right)^{s-2}\right)$ (Result from [2])
2-Paths in $n$ -node graph	$\mathcal{O}\left(\frac{2n}{q}\right)$ (Section 5.4.2)
Multiway Join: $N$ rels, $m$ vars., Dom. $n$ (Section 5.5.2)	Chain join: $(n/\sqrt{q})^{N-1}$ Star join: fact, dim. sizes $f, d_0: \frac{N d_0 (N d_0 / q)^{N-1}}{f + N d_0}$
$n \times n$ Matrix Multiplication	$\frac{2n^2}{q}$ for $q \geq 2n^2$ (Section 6.2 and [18])

Table 2: Representative upper bound on the replication rate  $r$  for each problem considered in this paper. This table only presents a representative upper bound, with a forward reference to the section that derives all upper bounds with constructive algorithms for each problem.

# Model

**LEMMA 3.1.** *For the Hamming-distance-1 problem, a reducer of size  $q$  can cover no more than  $(q/2) \log_2 q$  outputs.*


proof in technical report: F. N. Afrati, A. D. Sarma, S. Salihoglu, and J. D. Ullman. Upper and lower bounds on the cost of a map-reduce computation. CoRR, abs/1206.4377, 2012.

**THEOREM 3.2.** *For the Hamming-distance-1 problem with inputs of length  $b$ , the replication rate  $r$  is at least  $b / \log_2 q$ .*

# Hamming Distance 1

- **Deriving  $g(q)$ :** Recall that  $g(q)$  is the maximum number of outputs a reducer can cover with  $q$  inputs. By Lemma 3.1,  $g(q) = (q/2) \log_2 q$
- **Number of Inputs and Outputs:** There are  $2^b$  bitstrings of length  $b$ . The total number of outputs is  $(b/2)2^b$ . Therefore  $|I| = 2^b$  and  $|O| = (b/2)2^b$ .
- $\sum_{i=1}^p g(q_i) \geq |O|$  **Inequality:** Substituting for  $g(q_i)$  and  $|O|$  from above:

$$\sum_{i=1}^p \frac{q_i}{2} \log_2 q_i \geq \frac{b}{2} 2^b$$



$$r = \sum_{i=1}^p \frac{q_i}{2^b} \geq \frac{b}{\log_2 q}$$

# Hamming Distance 1

## Upper Bound: Splitting Algorithm

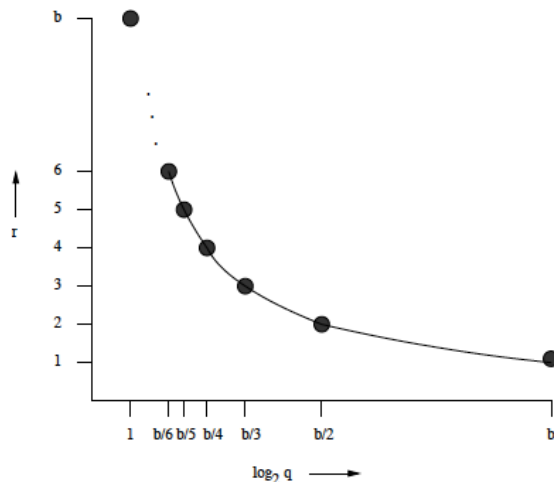


Figure 1: Known algorithms matching the lower bound replication rate

We can generalize the Splitting Algorithm so that for any  $c > 2$  such that  $c$  divides  $b$  evenly, we can have reducer size  $2^{b/c}$  and replication rate  $c$ . Note that for reducer size  $2^{b/c}$ , the lower bound on the replication rate is exactly  $b/\log_2(2^{b/c}) = c$ . We split each bit string  $w$  into  $c$  segments,  $w_1 w_2 \cdots w_c$ , each of length  $b/c$ . We will have  $c$  groups of reducers, numbered 1 through  $c$ . There will be  $2^{b-b/c}$  reducers in each group, corresponding to each of the  $2^{b-b/c}$  bit strings of length  $b - b/c$ . For  $i = 1, \dots, c$ , we map  $w$  to the Group- $i$  reducer that corresponds to bit string  $w_1 \cdots w_{i-1} w_{i+1} \cdots w_c$ , that is,  $w$  with the  $i$ th substring  $w_i$  removed. Thus, each input is sent to  $c$  reducers, one in each of the  $c$  groups, and the replication rate is  $c$ .

# Hamming Distance 1

## Upper Bound for large $q$ : Replicas on neighboring reducer

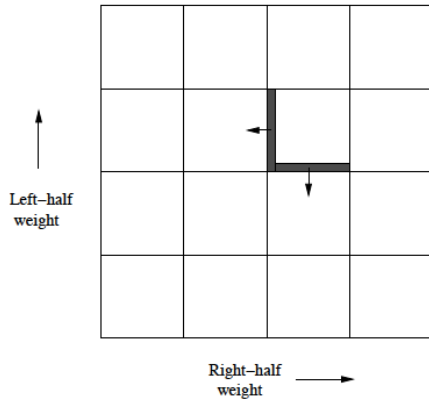


Figure 2: Partitioning by weight. Only the border weights need to be replicated

There is a family of algorithms that use reducers with large input –  $q$  well above  $2^{b/2}$ , but lower than  $2^b$ . The simplest version of these algorithms divides bit strings of length  $b$  into left and right halves of length  $b/2$  and organizes them by weights, as suggested by Fig. 2. The *weight* of a bit string is the number of 1's in that string. In detail, for some  $k$ , which we assume divides  $b/2$ , we partition the weights into  $b/(2k)$  groups, each with  $k$  consecutive weights. Thus, the first group is weights 0 through  $k - 1$ , the second is weights  $k$  through  $2k - 1$ , and so on. The last group has an extra weight  $b/2$ , and consists of weights  $\frac{b}{2} - k$  through  $b/2$ .

the replication rate is  $1 + \frac{2}{k}$ .

# Hamming Distance 1

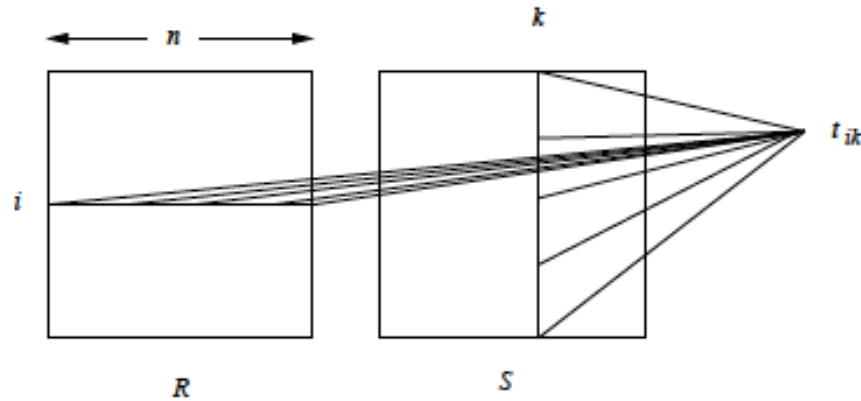


- Analysis for Hamming Distance 1 does not generalize easily to higher distance.
- Much higher bound for number of outputs covered by a reducer.

# Hamming Distance 1

- We are given a graph as input and want to find all triples of nodes such that in the graph there are edges between each pair of these three nodes.
- Alon Class of Sample Graphs: have the property that we can partition the nodes into disjoint sets, such that the subgraph induced by each partition is either:
  - A single edge between two nodes, or
  - Contains an odd-length Hamiltonian cycle.

# Triangle Finding



$$t_{i,k} = \sum_{j=1}^n r_{i,j} s_{j,k}$$

Figure 3: Input/output relationship for the matrix-multiplication problem

# Matrix Multiplication

## Matrix Multiplication Using Two Phases

1. In the first phase, we compute  $x_{ijk} = r_{ij}s_{jk}$  for each  $i, j$ , and  $k$  between 1 and  $n$ . We sum the  $x_{ijk}$ 's at a given reducer if they share common values of  $i$  and  $k$ , thus producing a *partial sum* for the pair  $(i, k)$ .
2. In the second phase, the partial sum for each pair  $(i, k)$  is sent from each reducer that has computed at least one  $x_{ijk}$  for some  $j$  to a reducer of the second phase whose responsibility to sum all these partial sums and thus compute  $t_{ik}$ .

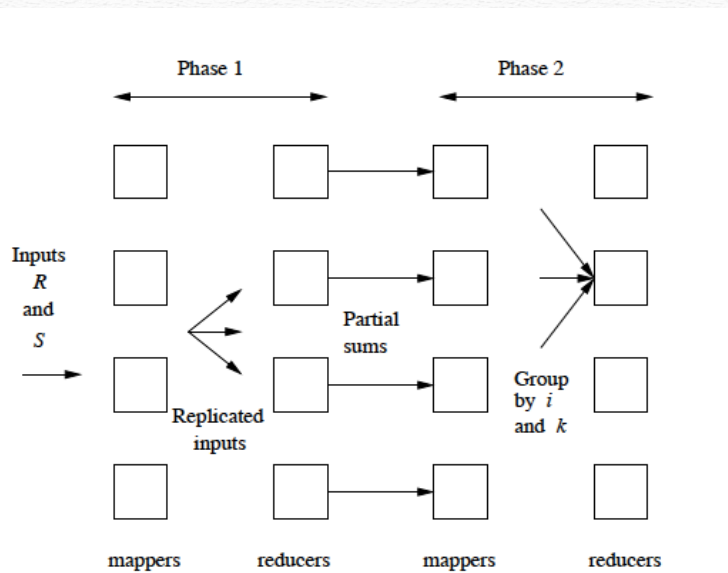


Figure 4: The two-phase method of matrix multiplication

# Matrix Multiplication

- <http://www.slideshare.net/tzulitai/upper-and-lower-bound-on-the-cost-of-a-map-reduce-computation>
- <http://shonan.nii.ac.jp/shonan/seminar011/files/2012/01/ullman.pdf>

# Reference

Q&A

**Thank you**

**22**