Communication Cost in Parallel Query Processing

Dan Suciu
University of Washington

Joint work with Paul Beame and Paris Koutris
Queries on Big Data

Traditional Data Processing:
• Main cost = disk I/O
• Postgres, Oracle, DB2, SQL Server…

Modern Large-scale Data Processing:
• Main cost = communication
• MapReduce, PigLatin, Spark, Shark,…
This Talk

• How much communication is needed to solve a “problem” on $p$ servers?
• This talk: define a parallel model (MPC) and analyze the communication cost for full conjunctive queries
• Why Highlights of Logic? Queries + parallel processing + communication complexity

References:
Beame, Koutris, S: *Skew in parallel query processing*, PODS 2014
Beame, Koutris, S: *Communication steps for parallel query processing*, PODS 2013
Outline

• The MPC Model

• Upper Bound for 1 round, no skew

• Lower Bound for 1 round, no skew

• Extensions/Discussions/Open Problems
Massively Parallel Communication Model (MPC)

extends BSP [Valiant]

Input data = size $M$ bits

Number of servers = $p$
Massively Parallel Communication Model (MPC)

- Input data = size $M$ bits
- Number of servers = $p$
- One round = Compute & communicate

extends BSP [Valiant]
Massively Parallel Communication Model (MPC)

extends BSP [Valiant]

Input data = size $M$ bits

Number of servers = $p$

One round = Compute & communicate

Algorithm = Several rounds

$O(M/p)$
Massively Parallel Communication Model (MPC)

extends BSP [Valiant]

Input data = size $M$ bits

Number of servers = $p$

One round = Compute & communicate

Algorithm = Several rounds

Max communication load per server = $L$
Massively Parallel Communication Model (MPC)

extends BSP [Valiant]

Input data = size $M$ bits

Number of servers = $p$

One round = Compute & communicate

Algorithm = Several rounds

Max communication load per server = $L$

Ideal: $L = M/p$; linear scaleup

This talk: $L = M/p^{1-\epsilon}$, $\epsilon \in [0,1]$; sublinear scaleup

Degenerate: $L = M$; $\epsilon = 1$; send data to one server

Most of this talk: 1 Round

...
Example: \( Q(x,y,z) = R(x,y) \bowtie S(y,z) \)

**Input:** \( R, S \)
- Uniformly partitioned on \( p \) servers

**Round 1:** each server
- Sends record \( R(x,y) \) to server \( h(y) \) mod \( p \)
- Sends record \( S(y,z) \) to server \( h(y) \) mod \( p \)

**Output:**
- Each server computes the local join \( R(x,y) \bowtie S(y,z) \)

**Assuming no skew**

1 Round
Load \( L = O(m/p) \) tuples
or \( O(M/p) \) bits

\[
\begin{array}{|c|c|}
\hline
x & y \\
\hline
a & b \\
\hline
a & c \\
\hline
b & c \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|}
\hline
y & z \\
\hline
b & d \\
\hline
b & e \\
\hline
c & e \\
\hline
\end{array}
\]
“No Skew”

- Expected load/server = \( \frac{m}{p} \)
- If \( \forall \) constant \( c \): 
  \[
  \deg_R(y=c) = O\left(\frac{m}{p (\log p)}\right)
  \]
  then the maximum load at all \( p \) servers = \( O\left(\frac{m}{p}\right) \) w.h.p. (Cernoff)
Summary of MPC Model

- **Data:** $m =$ # of tuples; $M =$ # of bits $= m \log m$
  - most of the talk assumes “no skew”

- **Servers:** $p =$ # of servers; powerful, randomized

- **Load:** $L =$ max # of bits/bytes received by one server

- **Problem:** compute a full conjunctive query* $Q$
  - [enumeration] problem (not [decision] problem)

- **Data complexity:** query $Q$ is fixed, $L =$ function of $m, p$

* without self-join (without loss of generality)
Outline

• The MPC Model

• Upper Bound for 1 round, no skew

• Lower Bound for 1 round, no skew

• Extensions/Discussions/Open Problems
Overview

Algorithm to compute a full conjunctive query in a single round of communication, by partially replicating the data

• The tradeoff was discussed [Ganguli’92]
• The algorithm first described by [Afrati&Ullman’10] for MapReduce
• Analysis and optimization [Beame’13,’14]: HyperCube Algorithm (this talk)
The Triangle Query

- **Input:** three tables
  \[ R(X, Y), \ S(Y, Z), \ T(Z, X) \]
  \[ |R| + |S| + |T| = m \text{ tuples} \]

- **Output:** compute
  \[ Q(x,y,z) = R(x,y) \bowtie S(y,z) \bowtie T(z,x) \]

- **Triangle** enumeration problem
  (not decision problem)

Quiz: what is the maximum # triangles?
Triangles in One Round

HyperCube Algorithm

• Place servers in a cube $p = p^{1/3} \times p^{1/3} \times p^{1/3}$

• Each server identified by $(i, j, k)$

$Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)$

$|R| + |S| + |T| = m$ tuples
**Triangles in One Round**

### Round 1:
- Send $R(x,y)$ to all servers $(h_1(x), h_2(y), *)$
- Send $S(y,z)$ to all servers $(*, h_2(y), h_3(z))$
- Send $T(z,x)$ to all servers $(h_1(x), *, h_3(z))$

### Output:
compute locally $R(x,y) \bowtie S(y,z) \bowtie T(z,x)$

\[
Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)
\]

\[|R| + |S| + |T| = m \text{ tuples}\]
Upper Bound is $L_{\text{algo}} = m/p^{2/3}$ tuples

**Theorem** Assume data has “no skew”. Then the algorithm computes $Q$ in one round, with communication load $L = O(L_{\text{algo}})$ w.h.p.
Theorem Assume data has “no skew”. Then the algorithm computes $Q$ in one round, with communication load $L = O(L_{\text{algo}})$ w.h.p.

- “No skew” means: all degrees $\leq O(m/p^{1/3})$
  If skew, then $L$ can grow up to $m/p^{1/2}$ (proof omitted)

- Sublinear scaleup $m/p^{2/3}$; is optimal when $|R| = |S| = |T|$

Proof. Correctness: each potential triangle $(x,y,z)$ is seen by server $(h_1(x), h_2(y), h_3(z))$
Load: expected load is $m/p^{2/3}$, hence so is the maximum load w.h.p.
HperCube Algorithm for Full CQ

\[ Q(x_1, \ldots, x_k) = S_1(\overline{x}_1), S_2(\overline{x}_2), \ldots, S_\ell(\overline{x}_\ell) \]

“Full”: all variables are in the head

• Write: \( p = p_1 \cdot p_2 \cdot \ldots \cdot p_k \)
  
  A server \( v \) has \( k \) coordinates \((v_1, \ldots, v_k)\)

• Round 1: send \( S_j(x_{j1}, x_{j2}, \ldots) \) to all \( v \) s.t.
  
  \[ h_{j1}(x_{j1}) = v_1, \quad h_{j2}(x_{j2}) = v_2, \quad \ldots \]

• Output: compute \( Q \) locally

\( p_i \) = the “share” of the variable \( x_i \)

\( h_1, \ldots, h_k \) = independent random hash functions
Computing Shares $p_1 \times p_2 \times \ldots \times p_k = p$

$|S_1| = m_1, |S_2| = m_2, \ldots, |S_\ell| = m_\ell$

Fix a relation $S_j(x_{j1}, x_{j2}, \ldots)$ with $m_j$ tuples
• Expected load/server $L_j = m_j / (p_{j1} \times p_{j2} \times \ldots)$
• Max load = $O(L_j)$ w.h.p. (because no skew)

[Afrati’10] nonlinear optimization:
$$L_{\text{algo}} = \min_{p_1, \ldots, p_k} \sum_j L_j$$

[Beame’13,14] linear optimization
$$L_{\text{algo}} = \min_{p_1,\ldots,p_k} \max_j L_j \quad (\text{next})$$
Computing Shares $p_1 \cdot p_2 \cdot \ldots \cdot p_k = p$

$L_{\text{algo}} = \min_{p_1, \ldots, p_k} \max_j L_j$

minimize $L$

$$\prod_i p_i \leq p$$

$\forall j: L \geq \frac{m_j}{\prod_{i:x_i \in S_j} p_i} = L_j$
Computing Shares $p_1 \times p_2 \times \ldots \times p_k = \mu_j = \log_p m_j, \lambda = \log_p L, e_i = \log_p p_i$

$L_{\text{algo}} = \min_{p_1, \ldots, p_k} \max_j L_j$

\[ \prod_i p_i \leq p \]

\[ \forall j : L \geq \frac{m_j}{\prod_{i: x_i \in S_j} p_i} = L_j \]

$L_{\text{algo}} = p^{\lambda^*}$

\[ \sum_i e_i \leq 1 \]

\[ \forall j : \lambda + \sum_{i: x_i \in S_j} e_i \geq \mu_j \]
Fractional Vertex Cover

Hyper-graph: nodes $x_1, x_2 \ldots$, edges $S_1, S_2, \ldots$

- **Vertex cover**: set of nodes $x_{i1}, x_{i2}, \ldots$ such that:
  \[ \forall S_j \exists x_{ih} \in S_j \]
- **Fractional vertex cover**: $v_1, v_2, \ldots v_k \geq 0$ s.t.:
  \[ \forall j : \sum_{i : x_i \in S_j} v_i \geq 1 \]

- **Fractional vertex cover value** is $\tau^* = \min_{v_1, \ldots, v_k} \sum_i v_i$

E.g. $Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)$

\[ \begin{array}{c}
  \text{1/2} \\
  1/2 \\
  \end{array} \]  \\
\[ \begin{array}{c}
  \text{1/2} \\
  \text{1/2} \\
  \end{array} \]

$\tau^* = 3/2$
Special Case: \( m_1 = m_2 = \ldots = m \)

- \( \sum_i e_i \leq 1 \) for all \( j \)
- \( \lambda + \sum_{i: x_i \in S_j} e_i \geq \mu \) independent on \( j \)

Optimal load is \( \lambda^* = \mu - 1/\tau^* \)

\[ L_{\text{algo}} = p^{\lambda^*} = m / p^{1/\tau^*} \]

Share exponents \( e_i \):
- \( e_1 = e_2 = e_3 = 1/3 \)
- \( p_1 = p_2 = p_3 = p^{1/3} \)
- \( L_{\text{algo}} = m/p^{2/3} \)

E.g. \( Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X) \) with \( \tau^* = 3/2 \)
Summary of HyperCube Algorithm

• Computes a full conjunctive query \( Q \) in one round

• No skew (will briefly discuss skew later)

• When all relations have equal size: \textit{fractional vertex cover} of \( Q \)
  
  Sub-linear scaleup: \( L_{\text{algo}} = m / p^{1/\tau} \)

• When relations have unequal sizes: \textit{fractional edge packing} of \( Q \)
  
  This follows from the lower bound analysis (next)
Outline

• The MPC Model

• Upper Bound for 1 round, no skew

• Lower Bound for 1 round, no skew

• Extensions/Discussions/Open Problems
Overview

• Will show: HyperCube is optimal for no-skew data, in a strong sense

• Proof techniques are interesting:
  – Friedgut’s inequality
  – Information theory
  – Use of the dual: fractional edge packing of $Q$
The Lower Bound for CQ

Fractional edge packing numbers \( u_1, u_2, \ldots, u_\ell \geq 0 \) s.t:

\[
\forall i : \sum_{j: x_i \in S_j} u_j \leq 1
\]

\[
L_{\text{lower}} = \max_{u_1, \ldots, u_\ell} \left( \frac{\prod_j M_j^{u_j}}{p} \right)^{1/\sum_j u_j}
\]

\( m = \) number of tuples, \( M = m \log(m) = \) number of bits

When \( M_1 = M_2 = \ldots = M \) then

\[
L_{\text{lower}} = \frac{M}{p^{1/\tau^*}}
\]

What comes next:
- Will give intuition
- Friedgut’s inequality
- Prove \( L_{\text{lower}} \) is lower bound, even for data with no skew

= value of fractional vertex cover (duality)
Hence, \( L_{\text{algo}} = L_{\text{lower}} \)
Intuition: Cartesian Product

Algorithm: [Ullman’2012] factorize \( p = p_1 \times p_2 \)
- Send \( S_1(x) \) to \((h(x),*)\) and \( S_2(y) \) to \((*,h(y))\)
- \( L = \max(\frac{m_1}{p_1}, \frac{m_2}{p_2}) \)
  Minimized when \( \frac{m_1}{p_1} = \frac{m_2}{p_2} = (\frac{m_1 m_2}{p})^{1/2} \)

Lower bound:
- A server receives \( \leq L \) tuples, Reports \( \leq L^2 \) pairs
- The \( p \) servers report all pairs:
  \( m_1 m_2 \leq p \ L^2 \)
Intuition: Cartesian Product

Consider any $Q$; the inputs $S_1, S_2, \ldots$ are on disjoint servers# Let $S_{j_1}, S_{j_2}, \ldots, S_{j_u}$ be an *edge packing* (i.e. no common vars)

**Claim** Any algorithm for $Q$ must also compute $S_{j_1} \times \ldots \times S_{j_u}$

**Proof** Since it doesn’t know the other relations

It follows $L_{\text{lower}} \geq \left( \frac{m_{j_1} \cdot m_{j_2} \cdots m_{j_u}}{p} \right)^{\frac{1}{u}}$

#otherwise, add another factor $|I|$
Friedgut’s Inequality [2004]

For any query $Q$ and fractional edge cover $u_1, u_2, \ldots$ of $Q$,

$$\sum_{x_1, x_2, \ldots, x_k \in [n]} a_1, x_1 \cdot a_2, x_2 \cdots a_\ell, x_\ell \leq \left( \sum_{x_1} a_1, x_1 \right)^{1/u_1} \cdot \left( \sum_{x_2} a_2, x_2 \right)^{1/u_2} \cdots \left( \sum_{x_\ell} a_\ell, x_\ell \right)^{1/u_\ell}$$

Cauchy-Schwartz: $Q(x) = S_1(x), S_2(x)$ \quad $u_1 = u_2 = 1/2$

$$\sum_x a_x b_x \leq \left( \sum_x a_x^2 \right)^{1/2} \times \left( \sum_x b_x^2 \right)^{1/2}$$

Generalized Holder: $Q(x) = S_1(x), S_2(x), S_3(x)$ \quad $u + v + w \geq 1$

$$\sum_x a_x b_x c_x \leq \left( \sum_x a_x \right)^{1/u} \times \left( \sum_x b_x \right)^{1/v} \times \left( \sum_x c_x \right)^{1/w}$$

Friedgut: $Q(x,y,z) = S_1(x,y), S_2(y,z), S_3(z,x)$ \quad $u_1 = u_2 = u_3 = 1/2$

$$\sum_{x,y,z} a_{xy} b_{yz} c_{zx} \leq \left( \sum_{x,y} a_{xy} \right)^{1/2} \times \left( \sum_{y,z} b_{yz} \right)^{1/2} \times \left( \sum_{z,x} c_{zx} \right)^{1/2}$$

Quiz: Friedgut’s proof: entropy. Try a direct, simple proof by induction on $k$.
Proof of Lower The Bound

\[ Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X) \]
size(R)+size(S)+size(T)=M

Denote \( L_{\text{lower}} = \frac{M}{p^{2/3}} \)

**Theorem** Any 1-round algorithm for \( Q \) has a load \( \geq L_{\text{lower}} \)

We prove a stronger claim. Suppose a deterministic algorithm for \( Q \) has load \( L_0 \)

**Lemma:** if \( L < L_{\text{lower}} \) then, over random permutations* \( R, S, T \), the algorithm \( A \) returns in expectation at most a fraction \( \left( \frac{L}{L_{\text{lower}}} \right)^{3/2} \) of \( Q \)‘s expected answers

Then use Yao’s lemma to extend to randomized algorithms

*Assume that the three input relations \( R, S, T \) are stored on disjoint servers; if not, extra factor 3
If \( L < L_{\text{lower}} \) then \( (L / L_{\text{lower}})^{3/2} \) fraction

\[
R, S, T = \text{random permutations on } [n] \\
|R| = |S| = |T| = n, \quad M = 3 \log n
\]

E.g. \( n = 4 \)

\[
\begin{array}{ll}
R & = \begin{pmatrix}
1 & 3 \\
2 & 1 \\
3 & 4 \\
4 & 2 \\
\end{pmatrix} \\
S & = \begin{pmatrix}
1 & 4 \\
2 & 2 \\
3 & 1 \\
4 & 3 \\
\end{pmatrix} \\
T & = \begin{pmatrix}
1 & 1 \\
2 & 2 \\
3 & 3 \\
4 & 4 \\
\end{pmatrix}
\end{array}
\]

\[
Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)
\]

\[
\text{size}(R) + \text{size}(S) + \text{size}(T) = M
\]

\[
L_{\text{lower}} = M / p^{2/3}
\]

Quiz: What is the minimum / maximum / expected \# triangles?
If $L < L_{\text{lower}}$ then $(L / L_{\text{lower}})^{3/2}$ fraction

Proof

$\Pr [(i,j) \in R] = 1/n$

same for $S, T$

One server $v \in [p]$: receives $L_R(v) + L_S(v) + L_T(v) \leq L$ bits

- Denote $a_{ij} = \Pr [(i,j) \in R \text{ and } v \text{ knows it (from the } L_R(v) \text{ bits about } R)]$
- similarly $b_{jk}, c_{ki}$ for $S$ and $T$

(1) $a_{ij} \leq 1/n$ (obvious) (2) $\Sigma_{i,j} a_{ij} \leq L_R(v) / \log n$ (formal proof uses entropy)

- $E [\#\text{answers to } Q] = \Sigma_{i,j,k} \Pr [(i,j,k) \in R \bowtie S \bowtie T] = n^3 * (1/n)^3 = 1$

$= \Sigma_{i,j,k} a_{ij} b_{jk} c_{ki} \leq [\Sigma_{i,j} a_{ij}^2] * [\Sigma_{j,k} b_{jk}^2] * [\Sigma_{k,i} c_{ki}^2]^{1/2}$ [Friedgut]

$\leq [L_R(v) / (n * \log n) * L_S(v) / (n * \log n) * L_T(v) / (n * \log n)]^{1/2}$ by (1), (2)

$\leq [L / (3 * n * \log n)]^{3/2}$ geometric/arithmetic mean

$= [L / M]^{3/2}$ size of $R, S, T$ is $M = 3 n \log n$

$E [\#\text{answers to } Q \text{ reported by all } p \text{ servers}] \leq p * [L / M]^{3/2} = [L / (M/p^{2/3})]^{3/2}$ QED
Discussion of the Lower Bound

• Lower bound under strong assumption: servers send \textit{bits}

• 1-round restriction is critical here:

\[
\Pr [(i,j,k) \in R \bowtie S \bowtie T \text{ and } v \text{ knows it}] = a_{ij} b_{jk} c_{ki}
\]

Fails for multiple rounds (weaker model there)

• An algorithm with load \( L = M / p^{1-\varepsilon} \), where \( \varepsilon < 1/3 \), reports only \( (L / L_{\text{lower}})^{3/2} = 1/p^{(1-3\varepsilon)/2} \) fraction of answers. Fewer, as \( p \) increases!
Outline

• The MPC Model

• Upper Bound for 1 round, no skew

• Lower Bound for 1 round, no skew

• Extensions/Discussions/Open Problems
Interpreting the Load Formula

\[ L_{\text{lower}} = \max_u \left( \frac{m_1^{u_1} \cdot m_2^{u_2} \cdots m_\ell^{u_\ell}}{p} \right)^{\frac{1}{\sum_j u_j}} \]

Fact 1 \( L_{\text{lower}} = \text{[geometric-mean of } m_1, m_2, \ldots \text{]} / p^{1/\Sigma u_j} \)

Fact 2 If \( m_j < L_{\text{lower}} \) then \( u_j = 0 \)
Proof: \( \lim_{u_j \to \infty} L(u_j) = m_j \), hence \( L(u_j) \) decreasing, max when \( u_j = 0 \)
Intuition: broadcast \( S_j \), remove from \( Q \)

Let \( S_k \) be the largest
If \( m_j \leq m_k / p \) (\( \leq L_{\text{algo}} \))
Then \( S_j \) is broadcast
Interpreting the Load Formula

**Fact 3** Optimal $u$ is a vertex of $Q$’s fractional edge packing polytope.

E.g. $Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)$

<table>
<thead>
<tr>
<th>Vertex of edge packing polytope</th>
<th>$(\frac{m_1 u_1 \cdot m_2 u_2 \cdot m_3 u_3}{p})^{\frac{1}{u_1+u_2+u_3}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1, u_2, u_3$</td>
<td>$(m_1 m_2 m_3)^{\frac{1}{3}} / p^{\frac{2}{3}}$</td>
</tr>
<tr>
<td>$1/2, 1/2, 1/2$</td>
<td>$(m_1 m_2 m_3)^{\frac{1}{3}} / p^{\frac{2}{3}}$</td>
</tr>
<tr>
<td>$1, 0, 0$</td>
<td>$m_1 / p$</td>
</tr>
<tr>
<td>$0, 1, 0$</td>
<td>$m_2 / p$</td>
</tr>
<tr>
<td>$0, 0, 1$</td>
<td>$m_3 / p$</td>
</tr>
</tbody>
</table>

$L_{\text{lower}}$ = the largest of these four values.
E.g. when $m_1=m_2=m_3=m$, then $L_{\text{lower}} = m / p^{2/3}$

**Fact 4** As $p$ increases, *scaleup rate* $\frac{1}{p^{1/\sum u_j}}$ decreases, up to $\frac{1}{p^{1/\tau^*}}$.

E.g. $m_1 = 8000$, $m_2=m_3=1000$:

- For $p < 64$: $L_{\text{lower}} = m_1 / p = 8000 / p$  
  linear scaleup
- For $p > 64$: $L_{\text{lower}} = (m_1 m_2 m_3)^{\frac{1}{3}} / p^{\frac{2}{3}} = 2000 / p^{\frac{2}{3}}$  
  sub-linear scaleup
\[ Q(x_1, \ldots, x_k) = S_1(\bar{x}_1), S_2(\bar{x}_2), \ldots, S_\ell(\bar{x}_\ell) \]

**AGM Bound**

[Atserias, Grohe, Marx’2008]; see excellent intuition in [Grohe’13]

For any **fractional edge cover** \( u_1, u_2, \ldots, u_\ell \)

\[ |Q| \leq m_1^{u_1} \cdot m_2^{u_2} \cdots m_\ell^{u_\ell} \]

When \( m_1 = m_2 = \ldots = m \) then \( |Q| \leq m^{\rho^*} \)

**Claim** An algorithm for \( Q \) on \( p \) servers has load \( L \geq m/p^{1/\rho^*} \)

**Proof.**

1. server outputs at most \( L^{\rho^*} \) tuples.

\( p \) servers output at most \( p \cdot L^{\rho^*} \geq m^{\rho^*} \) tuples.

E.g. join \( R(x,y), S(y,z) \)

- \( m/p^{1/\tau^*} \)
- \( m/p^{1/\rho^*} \)
- \( m/p^{1/2} \)

1 round

no skew

edge packing

Multi round

huge skew

edge cover

\[ \left( \frac{\prod_j M_j^{u_j}}{p} \right)^{\frac{1}{\sum_j u_j}} \]
Skew

**Definition** A constant $c$ is a “heavy hitter” for $x_i$ in $S_j$ if 
$\deg_{S_j}(x_i = c) > \frac{m_j}{p_i}$, where $p_i = \text{share of } x_i$

Two ways to handle skew:

- Unknown heavy hitters
- Known heavy hitters
Unknown Heavy Hitters

E.g. Join: \( Q(x,y,z) = R(x,y) \bowtie S(y,z) \), \(|R|=|S|=m\)

- **Standard hash-join algorithm:**
  - Shares: \( p = 1 \times p \times 1 \) (“hash on \( y \)”)
  - No skew: \( L = m/p \)
  - Worst skew: \( L = m \) -- bad...

- **HyperCube algorithm:**
  - Shares: \( p = p^{1/3} \times p^{1/3} \times p^{1/3} \)
  - No skew: \( L = m/p^{2/3} \)
  - Worst skew: \( L = m/p^{1/3} \) -- better!
Known Heavy Hitters

• There are at most $O(p)$ heavy hitters. Assume we know them, and we treat them specially. Then
  • For join: $Q(x, y, z) = R(x, y) \bowtie S(y, z)$, load is $L \leq m/p^{1/2}$
  • For triangles $Q(X, Y, Z) = R(X, Y) \bowtie S(Y, Z) \bowtie T(Z, X)$ load is $L \leq m/p^{1/2}$
• In both cases the exact load can be optimized based on the degrees of the heavy hitters
• Skew in a general query $Q$ is still ill understood

Open problem: better understanding of skewed data with known heavy hitters
Multiple Rounds

• [Beame’13] For no-skew data, we can reduce the load per server by allowing multiple rounds
  – Almost tight formula showing the tradeoff
  – Only for “tree-like” conjunctive queries
  – Lower bound uses weaker model of communication: servers send tuples

Open problem: lower bound for multi-rounds in the bit-model

Open problem: algorithms for multiple rounds on skewed data
Two Other Interesting Models

• MapReduce model [Afrati’13]: “Reducers” = “Servers”
  – They prove 1-round lower bounds for some CQ (subsumed by ours), and for similarity join
  – Weakness of MR: encourages large # of reducers $p$, which is bad for sub-linear scaleup $m / p^{1/r^*}$ (total communication = $m*p^{1-1/r^*}$)
    see story in [Ullman’12]

• DAG computation model and red/blue pebble games for I/O communication complexity:
  – Introduced by [Hong&Kung’81]
  – Lower bounds are for algorithms, not for problems (the bounds in this talk about about problems)
  – [Ballard’2012] proves a lower bound of $n^2 / p^{2/lg 7}$ for Strassen’s algorithm (includes multi rounds)
Thank You!