Complexity Theory for Map-Reduce

Communication and Computation Costs

Enumerating Triangles and Other Sample Graphs

Theory of Mapping Schemas

Coauthors

Foto Aftrati, Anish das Sarma, Dimitris Fotakis, David Menestrina, Aditya Parameswaran, Semih Salihoglu.

Cost of Map-Reduce Computations

- *Communication cost* = number of key-value pairs sent to reducers.
- *Computation cost* = execution time at reducers.
  - Computation at mappers normally proportional to communication cost.

Costs – Observations

- These costs are what you pay for at EC2.
- Often, communication cost dominates.
- Communication cost typically grows with the number of Reduce tasks.
- But latency shrinks with the number of tasks, so there is a tradeoff to be made.

Why One Round?

- "Other things being equal," it saves communication.
- But really: whatever you do with map-reduce, each round does something that you can study and perform as well as possible.

Finding All Instances of a Sample Graph

Communication Cost: Multiway Joins and Conjunctive Queries

Computation Cost: "Convertible Algorithms," Graph Decompositions

Triangles

- Given a data graph, find all triples of nodes that form a triangle.
- Use one round of map-reduce.
- Data graph represented by relation E(A,B).
  - A, B are nodes, and A<B (some order).
  - (A,B) is an edge.

Partition Method (Suri-Vassilvitskii)

- Partition nodes into b groups $S_1,\ldots,S_b$.
- Each reducer responsible for a set of three groups.
- Map to reducer $\{i,j,k\}$ all edges whose nodes are both in the union of S_i, S_j, S_k.
- Each reducer has a little graph – finds the triangles in that graph.

Partition Method – (2)

- An edge whose ends are in different groups is sent to (only) b-2 reducers.
- But an edge with both ends in the same group goes to {(b-1) choose 2} reducers.
- Communication cost (asymptotically) 3b/2 per edge.

Convention

- Data graph has n nodes and m edges; sample graph has p nodes.
  - p = 3 for triangle.

Our Approach

- Represent triangle-finding by a CQ
  E(X,Y) & E(X,Z) & E(Y,Z) & X<Y<Z.
- Use multiway join (Afrati & U, 2010).
- Hash nodes to b buckets.
- Reducer <-> list of buckets for X, Y, Z.
- Trick: < for nodes = bucket number.
  - Resolve ties by name of node.

Our Approach – (2)

- As a result, reducer [i,j,k] gets data only if $i \leq j \leq k$.
- Number of needed reducers = {(b+2) choose 3}, or approximately $b^3/6$.
- Each edge goes to exactly b reducers.
  - Which ones? Sort(node1, node2, any).
- Communication cost bm, vs. 3bm/2 (for the same number of reducers).

Generalization to All Sample Graphs

- For an arbitrary sample graph, we need one CQ for each order of the nodes.
  - p! CQ's, in principle.
- But the sample graph may have a nontrivial automorphism group.
- Example: square has 4! = 24 orders but 8 automorphisms.
  - Rotate to 4 positions, flip or don't.

Generalization – (2)

- We want only one CQ for each member of the quotient group (permutations/automorphisms).
- Example: square

E(W,X) & E(X,Y) & E(Y,Z) & E(W,Z) & W<X<Y<Z

E(W,X) & E(Y,X) & E(Y,Z) & E(W,Z) & W<Y<X<Z

E(W,X) & E(X,Y) & E(Z,Y) & E(W,Z) & W<X<Z<Y

Generalization – (3)

- Implement with one reducer for each nondecreasing sequence of p integers in the range [1, b] (number of buckets).
- That reducer gets all edges (i, j) if i<j and buckets of i and j are both in that sequence of integers.
- This reducer implements each of the conjunctive queries on its data.

Generalization – (4)

- Asymptotically $b^p/p!$ reducers.
- Asymptotically beats generalized partition (reducer <-> set of p blocks) by a small factor $1 + 1/(p-1)$.

Convertible Algorithms

- A serial algorithm is *convertible* (wrt a strategy for creating key-value pairs) if the total computation time of this algorithm at the reducers is of the same order as the serial algorithm.

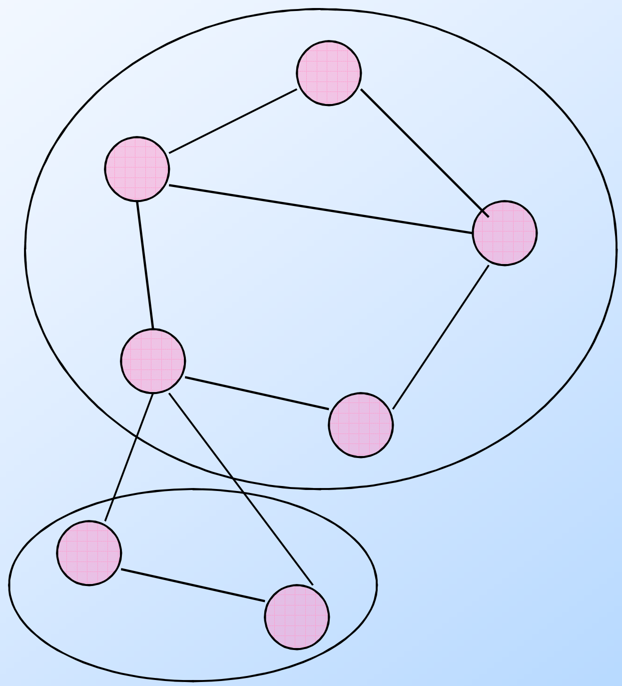
Convertible Algorithms – (2)

- Assuming random distribution of edges, a serial algorithm running in time $n^a m^b$ is convertible (with respect to partition or our scheme) iff $p \leq a + 2b$.
- For triangles, $O(m^{3/2})$ is achievable and best possible, so convertible.
  - $3 \leq 0 + 2(3/2)$.

Convertible Serial Algorithms

- There is an $O(m^{p/2})$ algorithm for many sample graphs.
  - Graphs with a Hamilton cycle.
  - Single edges.
  - Any combination of these.
    - Take union of graphs.
    - Throw in any additional edges you like.

Example



What If No Such Decomposition?

- If there are q isolated nodes after the best decomposition, then there is a serial algorithm with running time $O(n^q m^{(p-q)/2})$.
- All these algorithms are best possible (Noga Alon 1981).
  - They match the output size.
- All these algorithms are convertible.

Limited-Degree Data Graphs

- If there are no nodes of degree $\geq$ sqrt{m}, then for every connected sample graph there is a serial algorithm that runs in time $O(m^{p/2})$.
- Again – convertible.

Mapping Schemas

Definition

Examples: Triangles and Hamming Distance

A Lower Bound

Comments

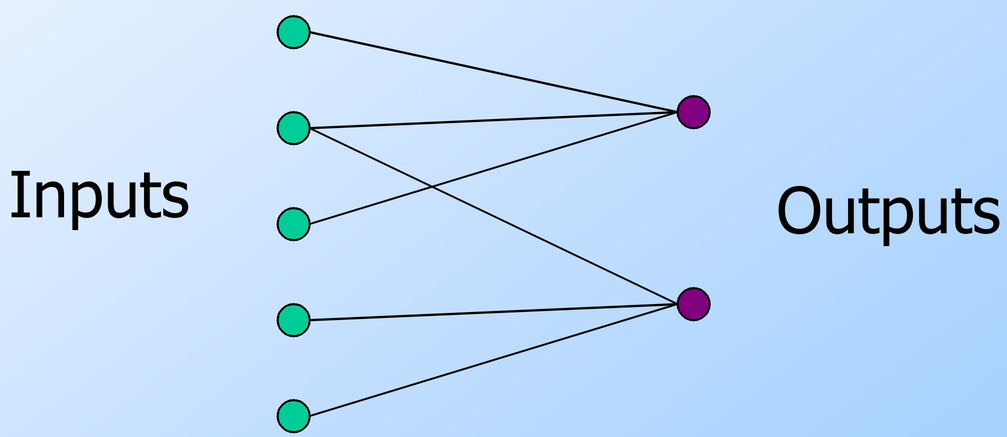
- Ideas are very new, not published or even written up.
- Approach originated with Anish das Sarma.
- We have results for finding sample graphs, Hamming distance, and containment join.
- We welcome work in this area.

Definition of Mapping Schema

- Set of inputs (that may be present, depending on the input data).
  - Distinction: for triangles, every possible edge is an "input"; some will really be there in any data set.
- Set of outputs.
- For each output: a set of inputs that must be present for that output to be made.

Example: Mapping Schema for Triangles

- Inputs = edges = pairs of nodes.
- Outputs = triangles = sets of three input edges that must be present for that triangle to be present in the graph.



Example: Mapping Schema for Hamming Distance = 1

- Inputs = binary strings of length b.
- Outputs = pairs of inputs of Hamming distance 1.

Mapping-Schema Optimization Problem

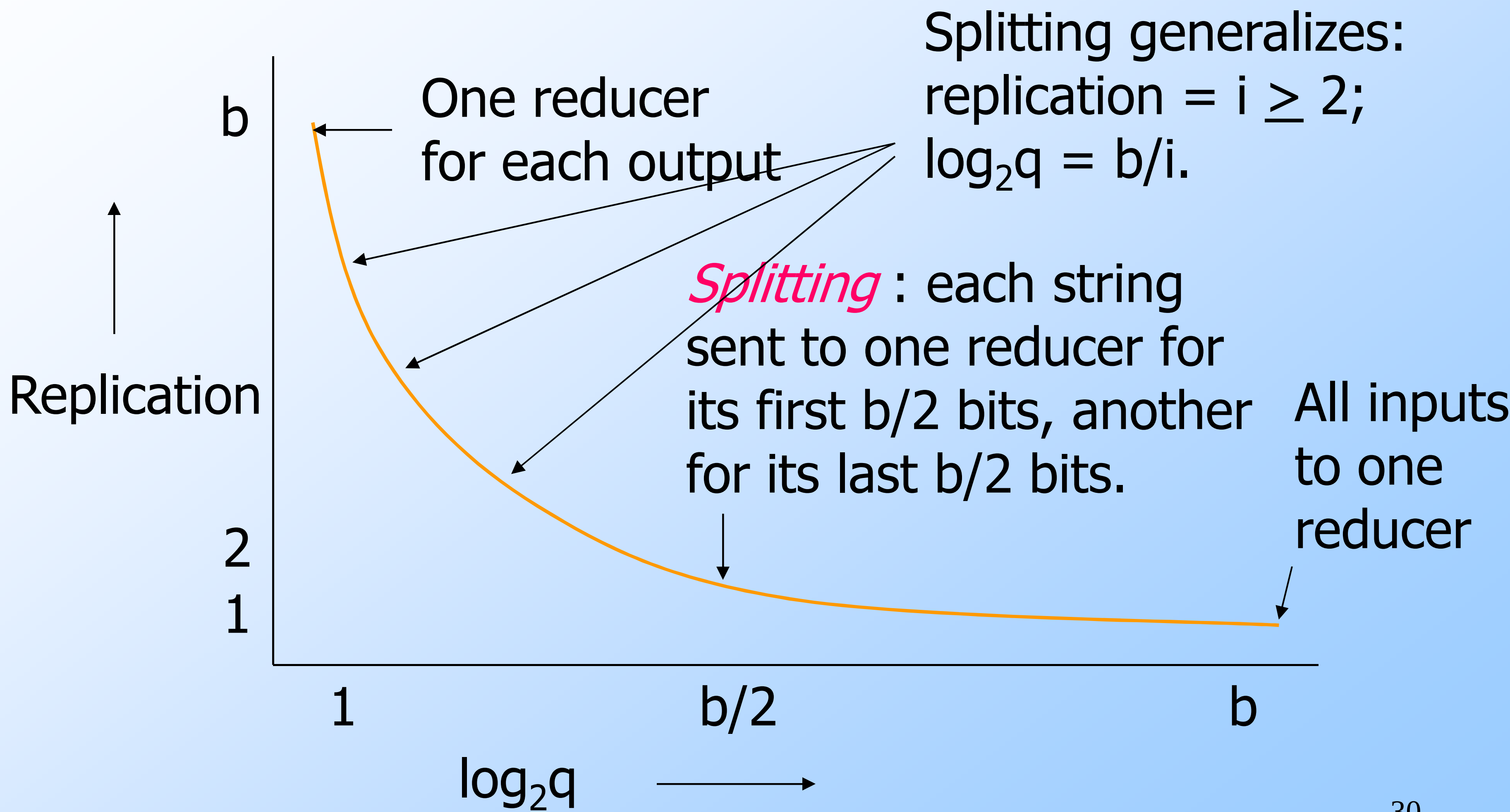
- Use p reducers.
- Each reducer assigned at most q inputs.
- For each output, its set of inputs must be contained in the set of inputs assigned to at least one reducer.
- Find input->reducer assignment to minimize *replication* = pq divided by the number of inputs.
  - = communication cost per input.

Lower Bound for HD =1

- Theorem (Semih Salihoglu): if a reducer gets q inputs, the maximum number of output sets it can cover is $(q/2) \log_2 q$.
- Since there are $(b/2)2^b$ outputs:

  $p(q/2) \log_2 q \geq (b/2)2^b$.
- Replication = $pq/2^b \geq b/\log_2 q$.

Communication/Computation Tradeoff



Research Program

1. Get upper/lower bounds on communication/reducer-size tradeoff for many different problems.
2. Relate structure of mapping schema to costs.
   - E.g., how does size of min-cuts relate to replication.