Matching Bounds for the All-Pairs MapReduce Problem

Foto Afrati[†], Jeffrey Ullman[‡]
[†]National Technical University of Athens, [‡]Stanford University

ABSTRACT

The all-pairs problem is an input-output relationship where each output corresponds to a pair of inputs, and each pair of inputs has a corresponding output. It models similarity joins where no simplification of the search for similar pairs, e.g., locality-sensitive hashing, is possible, and each input must be compared with every other input to determine those pairs that are "similar." When implemented by a MapReduce algorithm, there was a gap, a factor of 2, between the lower bound on necessary communication and the communication required by the best known algorithm. In this brief paper we show that the lower bound can essentially be met.

1. THE MAPREDUCE MODEL

We assume that the reader is familiar with MapReduce [3], and we use the model and definitions outlined in [1]. The essential terminology from the latter paper is:

1. A *mapper* is the application of the Map function to a single input.
2. A *reducer* is the application of the Reduce function to a single key and its associated list of values.
3. The *reducer size*, denoted q, is the maximum number of values that are allowed on the list for any reducer. The purpose of fixing q, and typically making it small is twofold. A small q allows the Reduce function to be executed in main memory, and it also forces there to be lots of parallelism available to the application.
4. A *problem* consists of a set of inputs, a set of outputs, and a relationship between those sets. For each output, the relationship tells us the set of inputs that are necessary to compute that output.
5. A *mapping schema* for a given problem and a given reducer size q is an assignment of inputs to sets of reducers, such that no reducer is assigned more than q inputs, yet for every output there is at least one reducer that receives all the inputs needed to produce that output.
6. The *replication rate* for a mapping schema is the average number of reducers to which each input is sent.

The results contained in [1] are upper and lower bounds on the replication rate r as a function of the reducer size q for a number of common problems. Upper bounds are the result of algorithms, while lower bounds are theorems saying replication rate below a certain function of q is impossible for a given problem. In all cases but one, the upper and lower bounds match, and it is this problem, which we call "the all-pairs problem," is the subject of this note. We show that the upper bound can be improved to match the lower bound.

2. THE ALL-PAIRS PROBLEM

[4] considers the design of a MapReduce algorithm for "drug interactions," where the inputs are large records about patients taking a given drug, and the outputs are those pairs of drugs that had a statistically significant probability of causing a heart attack when taken together. "Drug interactions" can be abstracted to a common input-output relationship, where the outputs are all possible pairs of inputs. This problem, called the *all-pairs problem*, was studied in [2] in its generalized form where inputs were of different sizes. Here, we revert to the simple form of problem where all inputs are assumed to be of unit size, and the reducer size is the number of inputs that can be sent to one reducer.

The lower-bound recipe given in [1] can be used to show that if there are d inputs, and the reducer size is q, then the replication rate r is lower bounded by $r \geq d/q$. However, [4] gives a mapping schema for the problem that has a replication rate of approximately twice the lower bound. In brief, that algorithm divides the d inputs into some number of groups, say g groups. For each pair of groups, there is one reducer that gets all the members of both groups. The replication rate is thus $r = g - 1$, and the reducer size is $q = 2d/g$, since each reducer gets the d/g members of two different groups. From these equations, we can derive $r = 2d/q - 1$, or approximately twice the lower bound d/q. The problem with this strategy is that at each reducer, about half the pairs of inputs are from the same group. Inputs from the same group are compared at $g - 1$ different reducers, so almost half the pairs of inputs at a reducer need not be compared there.

Here, we are going to show, by giving an improved mapping schema, that the lower bound can essentially be met. Our first step is to make the lower bound more precise, and a little higher, since the d/q bound is really just an asymptotic approximation to the true state of affairs.

THEOREM 2.1. *For the all-pairs problem*

$$r \geq \left\lceil \frac{d-1}{q-1} \right\rceil$$

PROOF. The technique of [2] can be used here. That is, consider any input. It needs to share a reducer with $d-1$ other inputs. But if it is sent by the mapping schema to a reducer, then there are at most $q-1$ other inputs sent to the same reducer. Thus, each input must be sent to at least $\lceil (d-1)/(q-1) \rceil$ reducers, and that is its minimum replication. □

3. THE MAPPING SCHEMA FOR THE ALL-PAIRS PROBLEM

Now, let us describe a mapping schema that essentially meets the lower bound of Theorem 2.1. Let p be a prime number. The problem with the algorithm described in [4] is that the groups are big, and since all members of a group are sent together to whatever reducers get that group, there is a lot of redundancy. Our solution is to use more groups, and smaller groups, and let each reducer take a different set of many groups. In that way, there is still redundancy because members of a group are always sent together to reducers, but at any reducer, the fraction of pairs from the same group is negligible.

The construction proceeds as follows. Divide the d inputs into p^2 equal-sized *groups*, each with d/p^2 inputs. We shall assume that p^2 divides d evenly, although if not, then the same algorithm works with slightly unequally sized groups. Think of the groups as arranged in a square, p on a side. The group in row i and column j is represented by (i, j). We shall assume i and j are each in the range 0 through $p-1$.

We'll use $p(p+1)$ reducers, organized as $p+1$ *teams* of p reducers in each team. In the kth team, $0 \leq k < p$, there is one reducer for each integer modulo p: $0, 1, \ldots, p-1$. In what follows, all arithmetic is performed modulo p, and we shall not say so explicitly. Send the inputs in group (i, j) to the reducer for $i + kj$ in each team k.

There is also a $(p+1)$st team, which we shall refer to as "team p." The reducers in team p correspond to the columns of the square. That is, the inputs in group (i, j) are sent to the reducer in team p that corresponds to j. Thus, every input is sent to $p+1$ reducers, and the replication rate is $p+1$.

We need to show that any two inputs are sent to some reducer in common. Let the two inputs be in groups (i, j) and (i', j'), respectively. Of course, if $i = i'$ and $j = j'$, i.e., the groups are the same, then many reducers get the members of the group, and any two inputs in the same group surely share a reducer. So let us consider the more interesting case where the two groups are not the same.

If $j = j'$, then the reducer for j in team p gets both. Otherwise, assume $j \neq j'$. In order for these two groups to go to the same reducer in team k, $0 \leq k < p$, it must be that $i + kj = i' + kj'$. That is, $(i - i') = k(j' - j)$. But we assume $j \neq j'$, so $(j - j')$ has an inverse modulo p. Then $k = (i - i')(j' - j)^{-1}$ is the value of k whose team has a reducer that gets both of these groups.

THEOREM 3.1. *If p is a prime, p^2 divides d, and $q = d/p$, then there is a mapping schema for the all-pairs problem with reducers of size q that has a replication rate $r = d/q + 1$.*

PROOF. We already noted that the mapping schema described above has $r = p+1$. The number of inputs in a group is d/p^2, and each reducer gets p groups. Thus, the necessary reducer size is $q = d/p$. Equivalently, $p = d/q$, so the theorem follows by substituting d/q for p in $r = p + 1$. □

Aside: You might notice that the lower bound

$$\left\lceil \frac{d-1}{q-1} \right\rceil$$

is not always less than the upper bound $d/q + 1$ from Theorem 3.1. However, the only condition under which

$$\frac{d-1}{q-1} > \frac{d}{q} + 1$$

is if $d > q^2$. In that case, p, which is d/q, is greater than $\sqrt{d}$, and it is not possible for p^2 to divide d. That is, there would be less than one input per group.

While the lower and upper bounds are not identical, we can show that there is no "daylight" between them. That is, the upper bound, which is always an integer, never exceeds $(d-1)/(q-1)$ by 1 or more, except in a trivial case. Suppose

$$\frac{d}{q} + 1 - \frac{d-1}{q-1} \geq 1$$

It follows that

$$\frac{d}{q} \geq \frac{d-1}{q-1}$$

This inequality can only hold if $q \geq d$. But in that case, one reducer suffices and $r = 1$ is both the upper and lower bound..

Since the lower bound $\lceil (d-1)/(q-1) \rceil$ is an integer, and the upper bound is also an integer for the values of d and q to which the upper bound applies, these integers must be the same. That is:

THEOREM 3.2. *For the values of d and q to which Theorem 3.1 applies, there is no algorithm with a lower replication rate.*

4. REFERENCES

[1] F. N. Afrati, A. Das Sarma, S. Salihoglu, and J. D. Ullman. Upper and lower bounds on the cost of a mapreduce computation. In *VLDB*, 2013.

[2] F. N. Afrati, S. Dolev, E. Korach, S. Sharma, and J. D. Ullman. *A Pseudopolynomial Solution for NP-hard Assignments in MapReduce*. Unpublished, Dept. of CS, Ben Gurion Univ. of the Negev, 2013.

[3] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. In *OSDI*, pages 137–150, 2004.

[4] J. D. Ullman. Designing good mapreduce algorithms. *XRDS*, 19(1):30–34, Sept. 2012.