Extensions and Complexity of MapReduce

Study of complexity is done to choose the best algorithms to solve a problem.

The communication cost complexity is the size of the input passed to the algorithm. As CPU speed is much greater than both network and disk speed, the majority of the time could be spent in communication.

For example,

For natural join

1. Mapper complexity = r + s
2. Reducer complexity = r + s
3. Overall = O(r + s)

For a three way join

1. Input to mapper 1 : r
2. Input to mapper 2 : s
3. Input to reducer 1 : r + s
4. For a probability p of a match between r and s, input to mapper 3 : prs
5. Input to mapper 4 : t
6. Input to reducer 2 : prs + t
7. Hence, overall = O(r + s + t + prs)
8. This depends on the order of join and we take the minimum of rs, ts and rt.

Wall clock time is said to be the time taken for the entire job to finish. Dividing jobs can increase communication time but decrease wall clock time, causing a tradeoff.

The parameters to study this tradeoff are

1. Reducer size q
    a. Max number of values that can have the same key - not the number of reducers
    b. If q is small, there can be more reducers
    c. This reduces wall clock time but increases communication time

2. Replication rate r
    a. R = number of key value pairs output by mapper / total number of input records to mapper

A group based algorithm is used to tradeoff between wall clock and communication time.

In this

1. Group data points
2. Read data in groups into a single reducer
3. Store this data in memory and compare all pairs from those groups
4. Output results

This way

1. There are g groups with m = n/g points
2. Each group is sent to g-1 servers
3. Total number of messages = g(g-1)
4. Total data = mg(g-1) = n(g-1) ~ ng
5. Parallelism is the number of nodes = $O(g^2) = {}^{n}C_{2}$.

This increases communication complexity but decreases wall clock time.