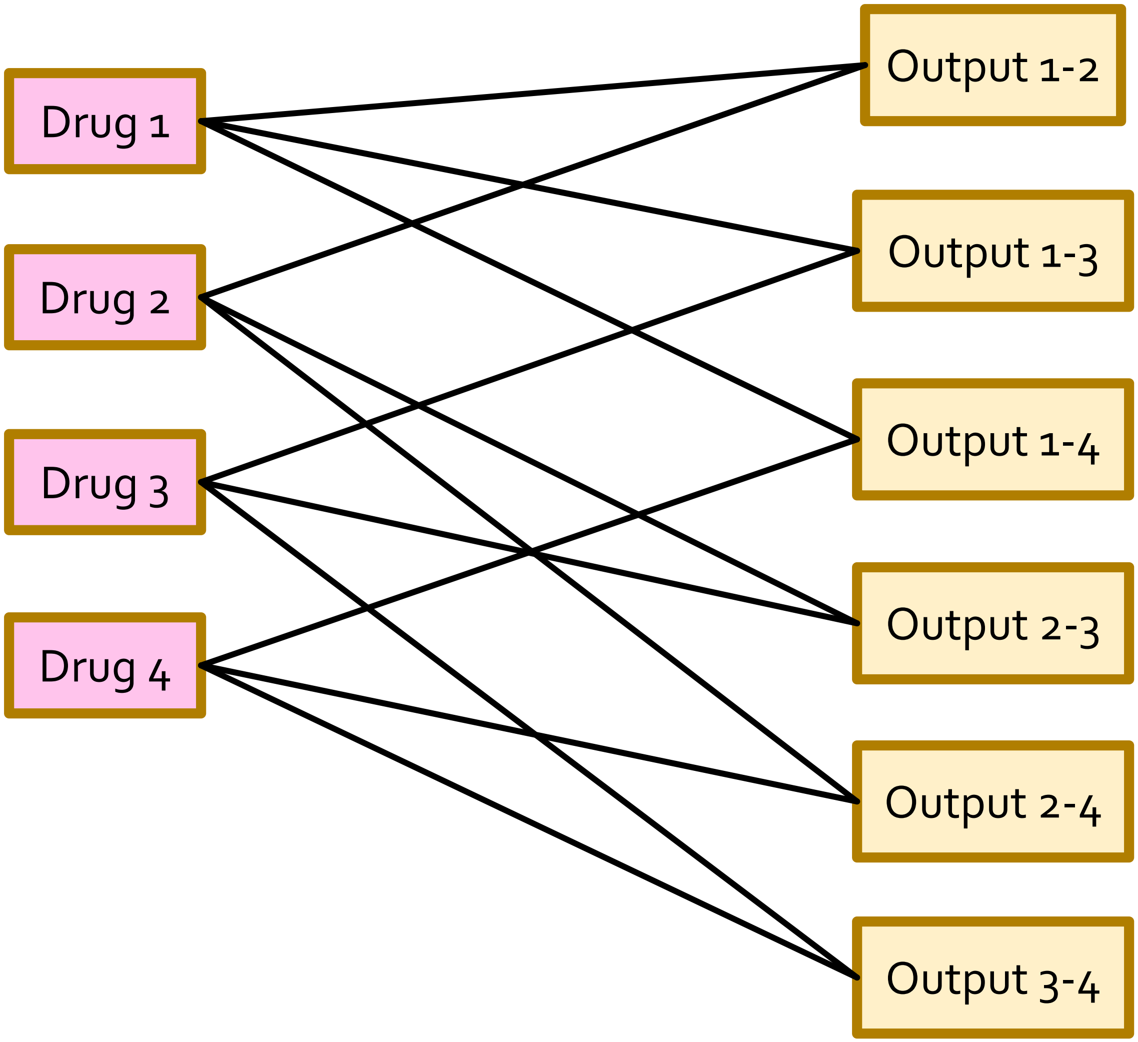
Theory of Map-Reduce Algorithms

Reducer Size
Replication Rate
Mapping Schemas
Lower Bounds

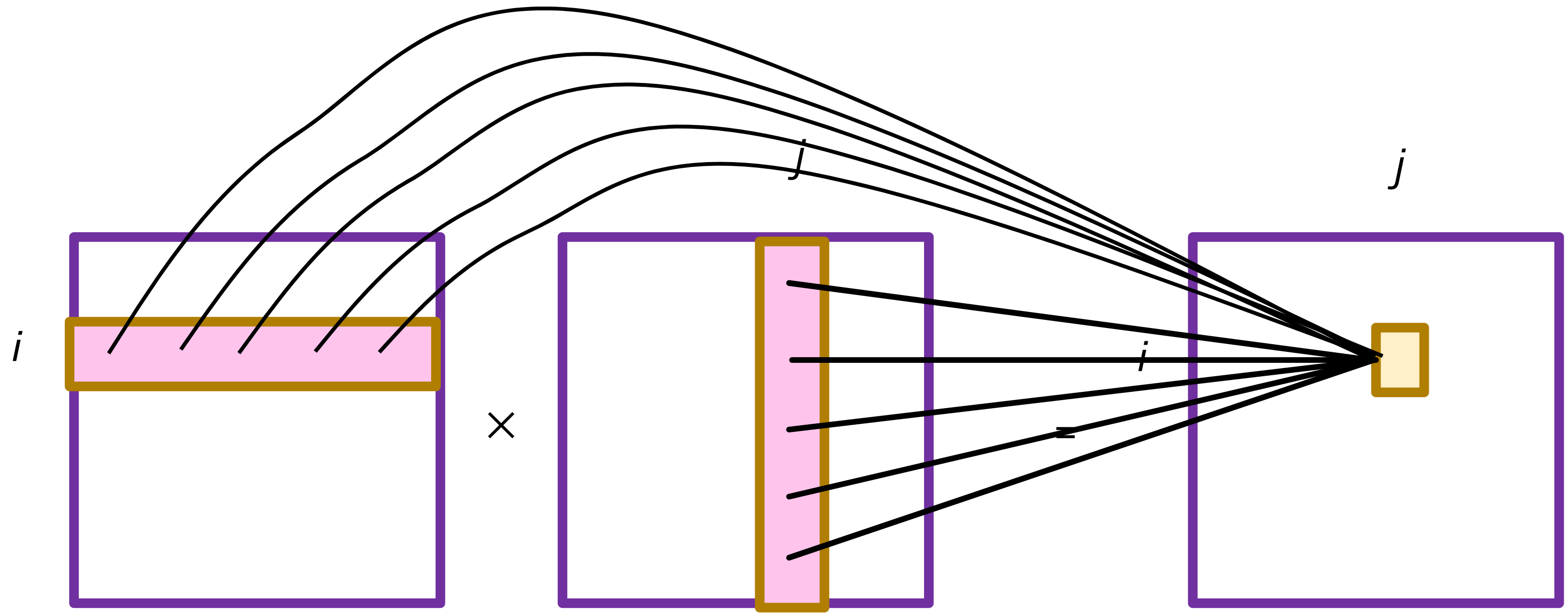
A Model for Map-Reduce Algorithms

1. A set of *inputs*.
   - Example: All drug numbers and their records.
2. A set of *outputs*.
   - Example: One output for each pair of drugs.
3. A many-many relationship between inputs and outputs.
   - An output is related to the inputs it needs to compute its values.
   - Example: The output for the pair of drugs {*i*, *j*} is related to inputs *i* and *j*.

Example: Drug Inputs/Outputs



Example: Matrix Multiplication



Hypothetical and Actual Inputs

- In our model, inputs and outputs are hypothetical.
  - That is, they represent all the possible inputs that could exist, and the outputs that might be made.
- In any execution of the algorithm, the real inputs are a subset of the hypothetical inputs, and the outputs are whatever can be made from those inputs.

Example: Join

- Let's consider how we would model the join of two relations R(A,B) and S(B,C).
  - Recall: this is the set of tuples (a,b,c) such that (a,b) is in R and (b,c) is in S.
- Inputs = all R(a,b) such that *a* and *b* are in the domains of attributes A and B, respectively + all S(b,c) such that *b* and *c* are in the domains of attributes B and C.
- Outputs = all (a,b,c) such that *a*, *b*, *c* are in the domains of A, B, C.

Example – Continued

- The input/output relationship connects each output (a,b,c) with two inputs: R(a,b) and S(b,c).
- In any actual instance of the problem, only a small fraction of the hypothetical inputs will be present in the relations R and S.
- An output can be made if and only if both of its related inputs are present.

Reducer Size

- Two parameters tell a great deal about a map-reduce algorithm: *reducer size* and *replication rate*.
- Reducer size, denoted q, is the maximum number of inputs that a given reducer can have.
  - I.e., the length of the value list.
- Limit might be based on how many inputs can be handled in main memory.

Scaling Up Reducer Size

- When not all hypothetical inputs are expected to be present, we can raise q proportionally.
- Example: If we want no more than one million inputs to any reducer, and we expect 10% of inputs to be present, then we can set q = 10M.
- Risk of *skew*.
  - Input selection may not be random.
  - Some reducers may get many more inputs than others.

Replication Rate

- The average number of key-value pairs created by each mapper.
  - Denoted r.
- Represents the communication cost per input.
- If each reducer has size q, there are p reducers, and I is the number of inputs for the problem, then $r = pq/I$.

Example: Drug Interaction

- Suppose we use g groups and d drugs.
- A reducer needs two groups, so $q = 2d/g$.
- Each of the d inputs is sent to $g-1$ reducers, or approximately $r = g$.
- Eliminate g to get $r = 2d/q$.
- Note: since d is I, the number of inputs, and p, the number of reducers is approximately $g^2/2$, the relationship $r = pq/I$ holds as well.
  - That is, $r = (g^2/2)(2d/g)/d = g$.

Upper and Lower Bounds on r

- What we did gives an upper bound on the least possible r as a function of q.
  - A particular algorithm is surely no better than the best possible algorithm.
- A solid investigation of map-reduce algorithms for a problem includes lower bounds.
  - Proofs that you cannot have lower r for a given q.

Mapping Schemas

- A *mapping schema* for a problem and a reducer size q is an assignment of inputs to sets of reducers, with two conditions:
  1. No reducer is assigned more than q inputs.
  2. For every output, there is some reducer that receives all of the inputs associated with that output.
     - Say the reducer *covers* the output.

Mapping Schemas – (2)

- Every map-reduce algorithm has a mapping schema.
- The requirement that there be a mapping schema is what distinguishes map-reduce algorithms from general parallel algorithms.

Example: Drug Interactions

- d drugs, reducer size q.
- No reducer can cover more than $q^2/2$ outputs.
- There are $d^2/2$ outputs that must be covered.
- Therefore, we need at least d^2/q^2 reducers.
- Each reducer gets q inputs, so replication r is at least $q(d^2/q^2)/d = d/q$.
- Half the r from the algorithm we described.
  - We had $r = g$ and $q = 2d/g$ (or $g = r = 2d/q$).