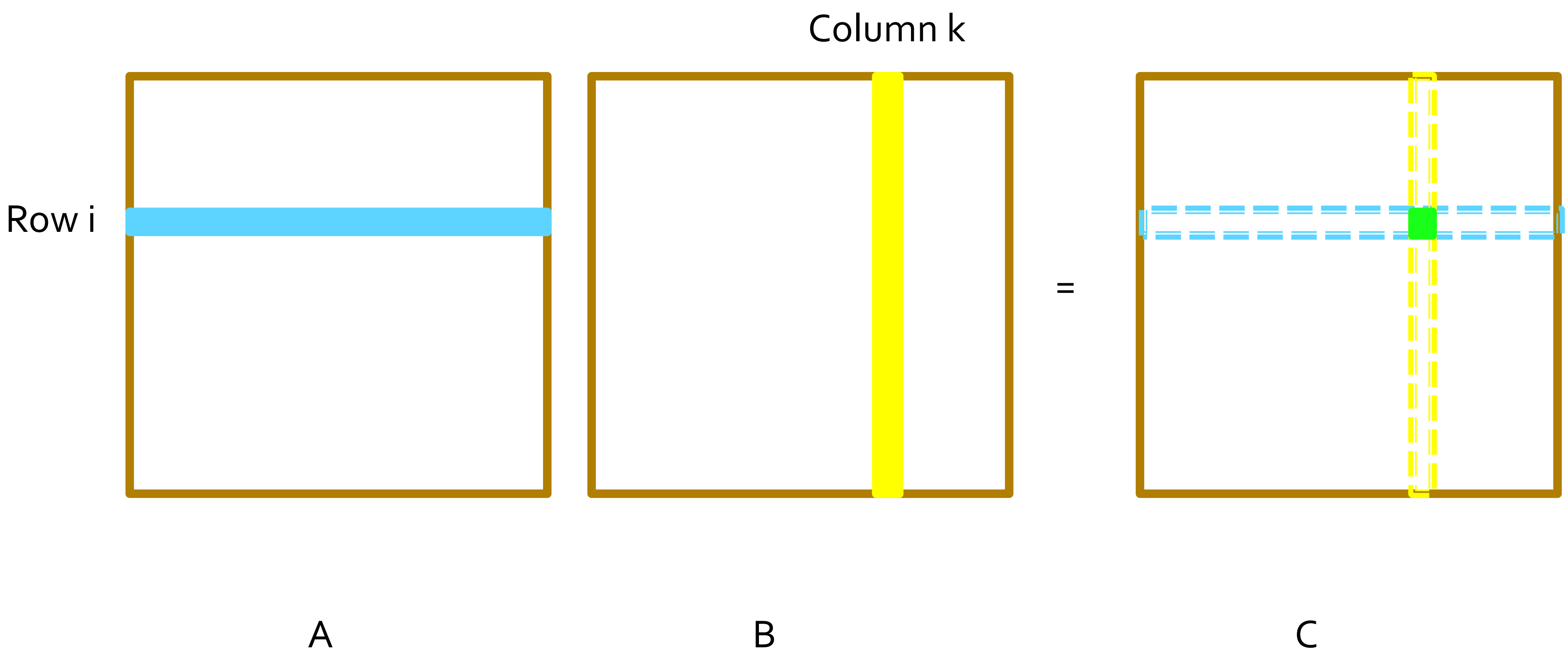
Matrix Multiplication

- One-Job Method
- Two-Job Method
- Comparison

Matrix Multiplication

- Assume n × n matrices AB = C.
- A_{ij} is the element in row *i* and column *j* of matrix A.
  - Similarly for B and C.
- $C_{ik} = \Sigma_j A_{ij} \times B_{jk}$.
- Output C_{ik} depends on the i^{th} row of A, that is, A_{ij} for all *j*, and the k^{th} column of B, that is, B_{jk} for all *j*.

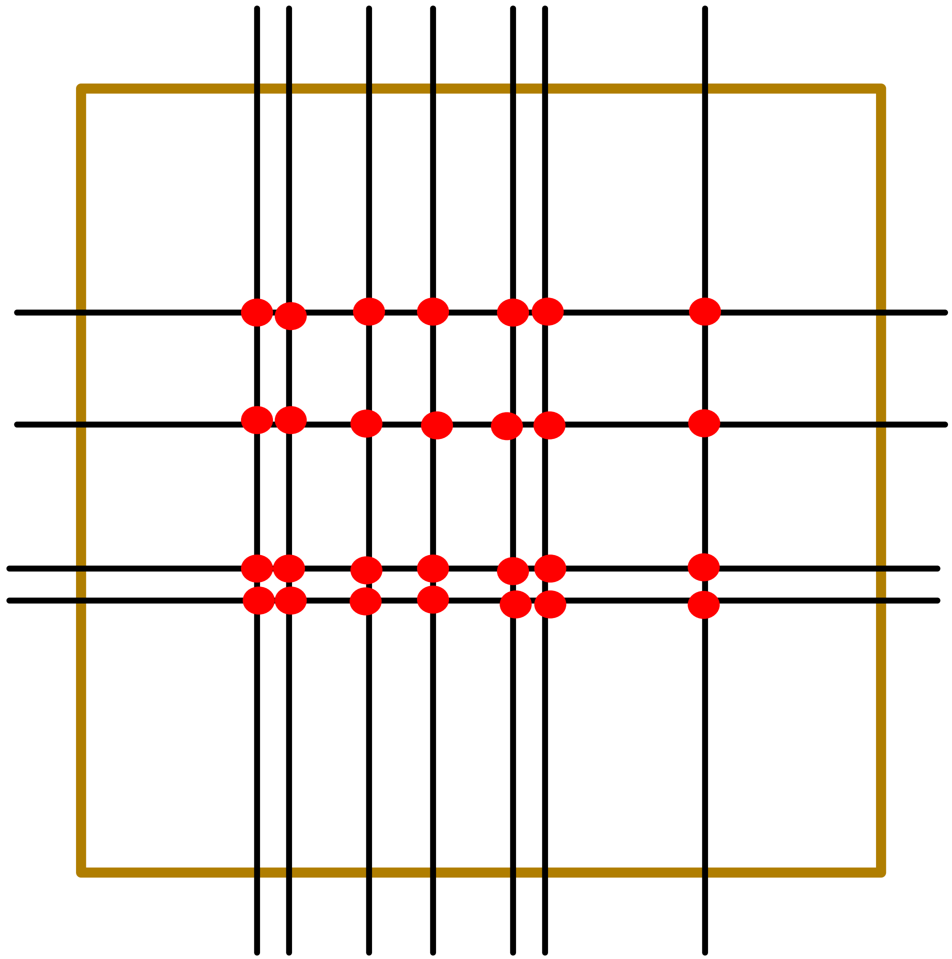
Computing One Output Value



Reducers Cover Rectangles

- Important fact: If a reducer covers outputs C_{ik} and C_{fg}, then it also covers C_{ig} and C_{fk}.
- Why? This reducer has all of rows *i* and *f* of A as inputs and also has all of columns *k* and *g* of B as inputs.
- Thus, it has all the inputs it needs to cover C_{ig} and C_{fk}.
- Generalizing: Each reducer covers all the outputs in the "rectangle" defined by a set of rows and a set of columns of matrix C.

The Responsibility of One Reducer



Upper Bound on Output Size

- If a reducer gets q inputs, it gets q/n rows or columns.
- Maximize the number of outputs covered by making the input "square."
  - I.e., #rows = #columns.
- q/2n rows and q/2n columns yield $q^2/4n^2$ outputs covered.

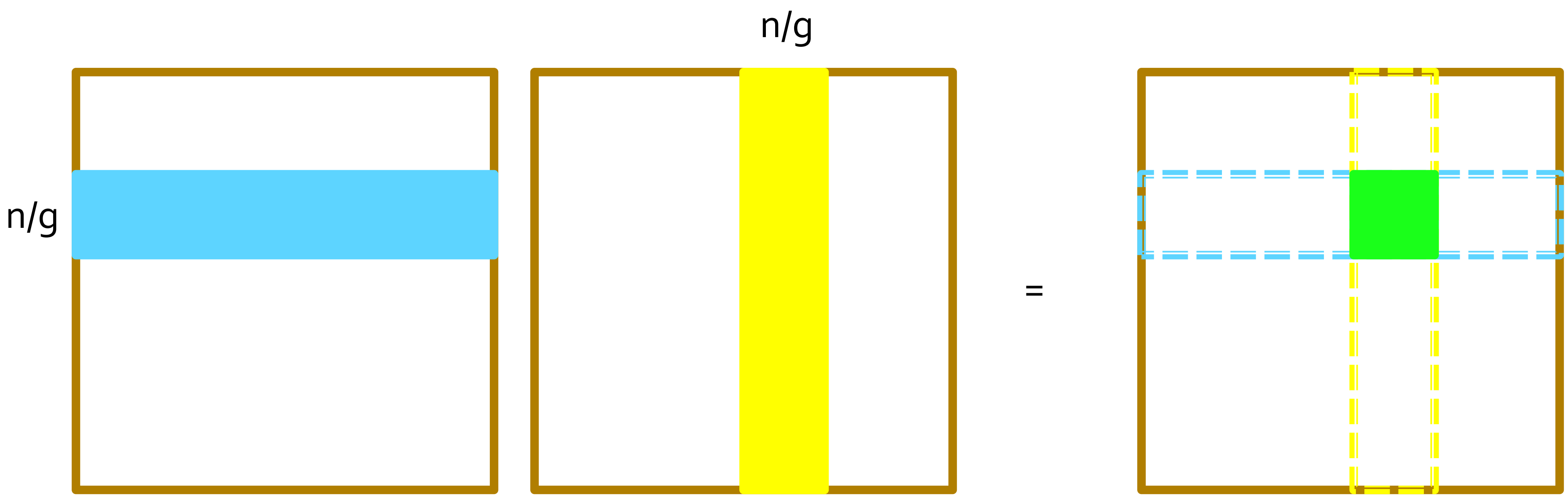
Lower Bound on Replication Rate

- Total outputs = n^2.
- One reducer can cover at most $q^2/4n^2$ outputs.
- Therefore, $4n^4/q^2$ reducers.
- $4n^4/q$ total inputs to all the reducers, divided by $2n^2$ total inputs = $2n^2/q$ replication rate.
- Example: If $q = 2n^2$, one reducer suffices and the replication rate is $r = 1$.
- Example: If $q = 2n$ (minimum possible), then $r = n$.

Matching Algorithm

- Divide rows of the first matrix into g groups of n/g rows each.
- Also divide the columns of the second matrix into g groups of n/g columns each.
- g^2 reducers, each with $q = 2n^2/g$ inputs consisting of a group of rows and a group of columns.
- $r = g = 2n^2/q$.

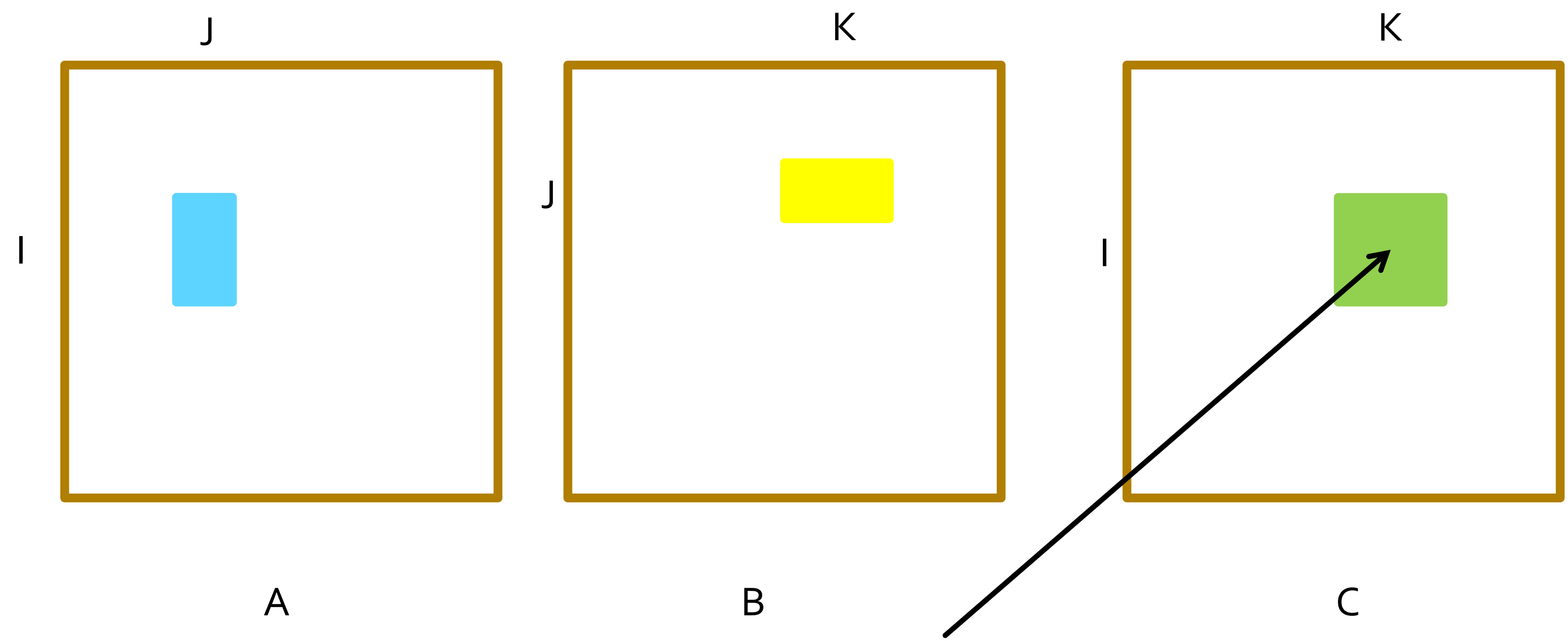
Picture of One Reducer



Two-Job Map-Reduce Algorithm

- A better way: use two map-reduce jobs.
- Job 1: Divide both input matrices into rectangles.
  - Reducer takes two rectangles and produces partial sums of certain outputs.
- Job 2: Sum the partial sums.

Picture of First Job



For i in I and k in K, contribution is $\Sigma_{j \text{ in } J} A_{ij} \times B_{jk}$

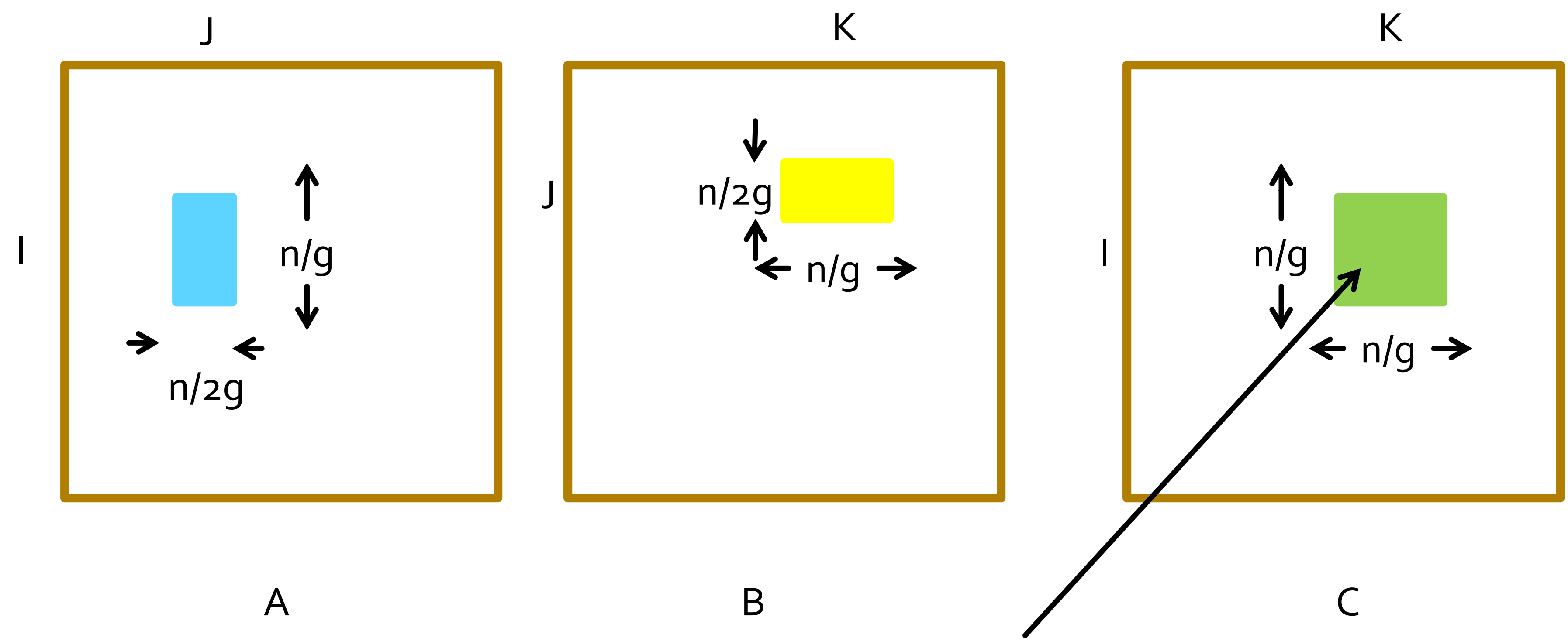
First Job – Details

- Divide the rows of the first matrix A into g groups of n/g rows each.
- Divide the columns of A into 2g groups of n/2g.
- Divide the rows of the second matrix B into 2g groups of n/2g rows each.
- Divide the columns of B into g groups of n/g.
- Important point: the groups of columns for A and rows for B must have indices that match.

Reducers for First Job

- Reducers correspond to an n/g by n/2g rectangle in A (with row indices I, column indices J) and an n/2g by n/g rectangle in B (with row indices J and column indices K).
  - Call this reducer (I,J,K).
  - Important point: there is one set of indices J that plays two roles.
    - Needed so only rectangles that need to be multiplied are given a reducer.

The Reducer (I,J,K)



2g reducers contribute to this area, one for each J.

Job 1: Details

- Convention: i, j, k are individual rows and/or column numbers, which are members of groups I, J, and K, respectively.
- Mappers Job 1:
  - A_{ij} -> key = (I,J,K) for any group K; value = (A,i,j,A_{ij}).
  - B_{jk} -> key = (I,J,K) for any group I; value = (B,j,k,B_{jk}).
- Reducers Job 1: For key (I,J,K) produce $x_{iJk} = \Sigma_{j \text{ in } J} A_{ij} \times B_{jk}$.

Job 2: Details

- Mappers Job 2: x_{iJk} -> key = (i,k), value = x_{iJk}.
- Reducers Job 2: For key (i,k), produce output $C_{ik} = \sum_J x_{iJk}$.

Comparison: Computation Cost

- The two methods (one or two map-reduce jobs) essentially do the same computation.
  - Every A_{ij} is multiplied once with every B_{jk}.
  - All terms in the sum for C_{ik} are added together somewhere, only once.
- 2 jobs requires some extra overhead of task management.

Comparison: Communication Cost

- One-job method: $r = 2n^2/q$; there are $2n^2$ inputs, so total communication $= 4n^4/q$.
- Two-job method with parameter g:
  - Job 2: Communication $= (2g)(n^2/g^2)(g^2) = 2n^2g$.

Number of reducers contributing to each output

Number of output squares

Area of each square

Communication Cost – Continued

- Job 1 communication:
  - $2n^2$ input elements.
  - Each generates g key-value pairs.
  - So another $2n^2g$.
  - Total communication = $4n^2g$.
- Reducer size $q = (2)(n^2/2g^2) = n^2/g^2$.
  - So $g = n/\sqrt{q}$.
  - Total communication = $4n^3/\sqrt{q}$.
    - Compares favorably with $4n^4/q$ for the one-job approach.