CSE 4125: Distributed Database Systems
Chapter – 6

Optimization of Access Strategies.

(part – A)

Outline

- Query optimization
- Problems in query optimization

Query Optimization*

- Permutation of the ordering of operations within a query can provide many equivalent **strategies** to execute it.
- Finding an "**optimal**" ordering of operations for a given query is important.
  - Done by query optimization layer(or optimizer for short).

Challenges in Query Optimization

- Materialization
  - Data (physical images) on which the query is executed.
- Order of execution
  - i.e. good sequence of join, semi-join, union etc.
- Method of execution
  - i.e. sequentially/ parallel, clustering the operations etc.

Objective of Query Optimization*

The selection of the optimal strategy consists–

- ***a search space:*** the set of alternative execution plans that represent the input query.
- ***a cost model:*** predicts the cost of a given execution plan.
- ***a search strategy:*** explores the search space and selects the best plan, using the cost model.

Objective of Query Optimization* (contd.)

Cost model needs to measure –

1. **Execution cost:** weighted combination of I/O, CPU, and communication costs.
2. **Fragment statistics (Database profiles):**
   – Estimating the amount of data in the fragments.
   – Estimating the cardinalities of results of relational operations.

Performance Measurement

- The selection of the optimal strategy is made by measuring their expected performances.
- **In centralized DB,**
  - Number of I/O operations
  - Usage of CPU

Performance Measurement (contd.)

- The selection of the optimal strategy is made by measuring their expected performances.
- **In DDB,**
  - Number of I/O operations
  - Usage of CPU
  - Data transmission requirements *[dominant]*

Data Transmission

Data transmission requirement can be evaluated by –

- **Transmission cost**
  - i.e. cost to initiate a transmission, routing cost etc.
- **Transmission delay**
  - i.e. elapse time between activation and completion of an app.

Data Transmission (contd.)

Data transmission requirement can be evaluated by –

- **Transmission cost**

  $TC(x) = C_0 + x * C_1$

- **Transmission delay**

  $TD(x) = D_0 + x * D_1$

Data Transmission (contd.)

Data transmission requirement can be evaluated by –

- **Transmission cost**

  $TC(x) = C_0 + x * C_1$

- **Transmission delay**

  $TD(x) = D_0 + x * D_1$

x = Transmitted data

C's and D's are system dependent constants.

C_0 = initialization fixed cost
C_1 = network wide unit cost
D_0 = connection initialization fixed time
D_1 = network wide unit transfer rate

Data Transmission (contd.)

Data transmission requirement can be evaluated by *(more detailed characterization)*

- Transmission cost

  $TC(x) = C_0^{ij} + x * C_1^{ij}$

- Transmission delay

  $TD(x) = D_0^{ij} + x * D_1^{ij}$

i and *j* denote source and destination respectively.

Database Profiles

What are Database Profiles*?

- Statistical information of the database.
- Necessity:
  - To perform sequence of operations, relations must be transmitted over the network.
  - It is important to estimate the size of the results to minimize the data transfers.
  - We need the statistical information (**database profile**) to estimate.

Information in Database Profiles

For a relation $R(A_1, A_2, ..., A_n)$ with fragments $R_1, R_2, ..., R_r$ the database profile contains following information.

- **card (R_i):** number of tuples of R_i.
- **size (A_i):** size or length (i.e. number of bytes) of attribute A_i.
  - For simplicity, same attribute name → same size
- **size (R_i):** sum of the size of all attributes of R_i.

Information in Database Profiles (contd.)

For a relation $R(A_1, A_2, ..., A_n)$ with fragments $R_1, R_2, ..., R_r$ the database profile contains following information.

- **val (A_i [R_i]):** number of distinct values appearing for attribute A_i of R_i.
- **dom(A_i):** domain of an attribute.
- **site (R_i)**: allocated site of the fragment R_i.

Database Profiles (example)

card ($DEPT_1$) = 10
site($DEPT_1$) = 2

	deptnum	name	area	mgrnum
size	2	15	1	7
val	10	10	2	10

Q: size (R_i) = ?

Estimating profiles of results of algebraic operations

What to Estimate?

- Estimating the profiles of results of algebraic operations.
- This information is useful for optimization (previous slides).
- Assume, *R* and *S* are input fragments and *T* is the result.
  - We will mostly estimate *card(T)* and *size(T)*.
  - Example: If *card(R)* and *card(S)* is given, can we estimate *card(T)* for *T = R* UN *S*?

Union

T = *R* UN *S*

card (T) ? card (R) ? card (S)

size (T) ? size (R) ? size (S)

Selection

$T = SL_{A = value}\, R$

card (T) = ρ * card (R)

Here ρ is selectivity.

$$\rho = \frac{1}{val\,(\,A[R]\,)}$$

Selection (contd.)

$T = SL_{A = value}\ R$

card (T) = ρ * card (R)

Here ρ is selectivity.

$$\rho = \frac{1}{val\,(\,A[R]\,)}$$

example

T = SL $_{A1 = B}$ **R**

ρ = ?

card(T) = ?

R

A1	*A2*
A	E
B	F
C	G
D	H

Selection (contd.)

$T = SL_{A\ =\ value}\ R$

card (T) = ρ * card (R)

Here ρ is selectivity.

$$\rho = \frac{1}{val\ (\ A[R]\ )}$$

example

$\mathbf{T} = SL_{A1\ =\ B}\ \mathbf{R}$

ρ = ?

card(T) = ?

R

A1	*A2*
A	E
B	F
A	G
B	H

Selection (contd.)

$T = SL_{A = value}\ R$

card (T) = ρ * card (R)

Here ρ is selectivity.

$$\rho = \frac{1}{val\,(\,A[R]\,)}$$

example

$\mathbf{T} = SL_{A1 = B}\ \mathbf{R}$

ρ = ?

card(T) = ?

R

A1	*A2*
A	E
B	F
C	G
B	H

Selection (contd.)

$T = SL_{A = value} R$

card (T) = ρ * card (R)

Here ρ is selectivity.

$$\rho = \frac{1}{val\,(\,A[R]\,)}$$

example

T = SL $_{A1 = B}$ **R**

ρ = ?

card(T) = ?

R

A1	*A2*
A	E
A	F
A	G
B	H

Selection (contd.)

$T = SL_{A\,=\,value}\ R$

card (T) = ρ * card (R)

Here ρ is selectivity.

$$\rho = \frac{1}{val\,(\,A[R]\,)}$$

Assuming, values are homogeneously distributed

example

$\mathbf{T} = SL_{A1\,=\,B}\ \mathbf{R}$

ρ = ?

card(T) = ?

R

A1	*A2*
A	E
A	F
A	G
B	H

Selection (contd.)

According to *Selinger et al.* (1979),

- For $A > value$,

$$\rho = \frac{\max(A) - value}{\max(A) - \min(A)}$$

- For $A < value$,

$$\rho = \frac{value - \min(A)}{\max(A) - \min(A)}$$

Selection (contd.)

size (T) = ?

Projection

$\boldsymbol{T} = PJ_A\ \boldsymbol{R}$

card (T) = ?

example

R

A1	*A2*
A	E
B	F
C	G
B	H

Projection (contd.)

$\boldsymbol{T} = \mathrm{PJ}_A\ \boldsymbol{R}$

card (T) = val (A[R])

Projection (contd.)

$\boldsymbol{T} = PJ_A \boldsymbol{R}$

size (T) ? size (R)

Cartesian Product

T* = *R* CP *S

card (T) = card (R) × card (S)

Join

$$T = R \text{ JN}_{R.A = S.B} \; S$$

card (T) = ?

Join (contd.)

$$T = R \ JN_{R.A = S.B} \ S$$

card (T) = selectivity * card(R *CP* S)

= ρ * (card(R) × card (S))

Join (contd.)

$$T = R \text{ JN}_{R.A = S.B} \; S$$

card (T) = selectivity * card(R *CP* S)

$$= \rho * (\text{card}(R) \times \text{card}(S))$$

$$= \frac{1}{\text{val}(A[R])} \times \text{card}(R) \times \text{card}(S)$$

$$= \frac{\text{card}(R) \times \text{card}(S)}{\text{val}(A[R])} = \frac{\text{card}(R) \times \text{card}(S)}{\text{val}(B[S])}$$

Semi-Join

$$T = R \; SJ_{R.A = S.B} \; S$$

card (T) = ?

Semi-Join (contd.)

$T = R \; SJ_{R.A = S.B} \; S$

card (T) = ρ * card(R)

According to *Hevner et al. (1979),*

$$\rho = \frac{val\,(A[R])}{val\,(dom(A))}$$

Optimization Graph

Optimization Graph

- A model to describe query optimization.
- Convenient than operator tree.
- Include only *critical* operations (critical for data transmission)

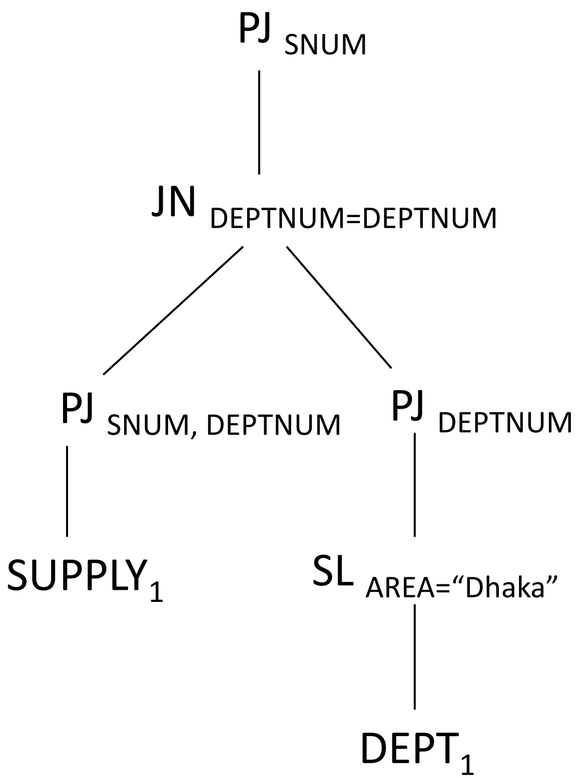
Optimization Graph (contd.)

- Unary operations are *not critical*.
  - Effect only by reducing operands and do not need data transmission.
  - These operations are collected by a program called ***fragment reducer***.

Optimization Graph (contd.)

- Unary operations are *not critical*.
  - Effect only by reducing operands and do not need data transmission.
  - These operations are collected by a program called ***fragment reducer***.
- Binary operations are *critical*.
  - When operands are not in the same site, they need data transmission.
  - CP, DF and SJ are not considered as they are rare. JN and UN are kept which gives us a graph called **optimization graph**.

Optimization Graph (example)



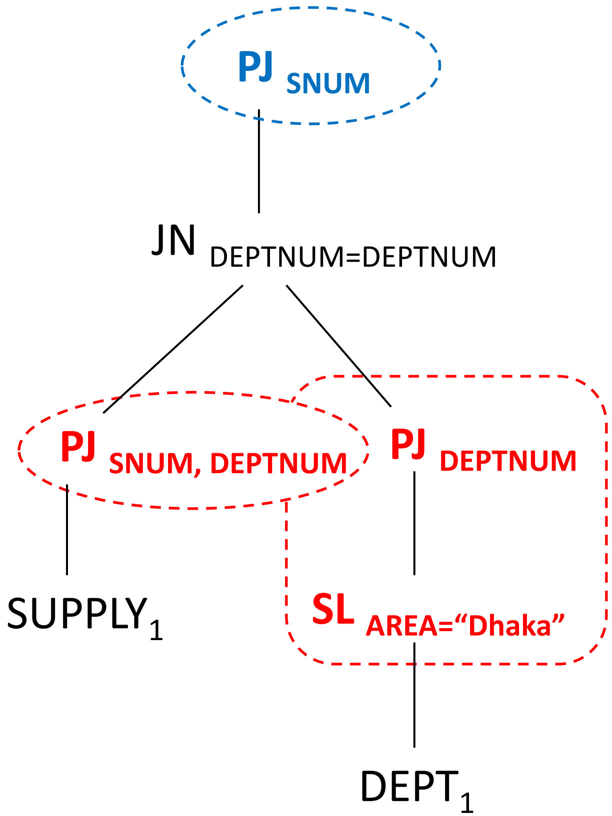
card ($SUPPLY_1$) = 30000
site($SUPPLY_1$) = 1

	snum	pnum	deptnum	quan
size	6	7	2	10
val	1800	1000	20	500

card ($DEPT_1$) = 10
site($DEPT_1$) = 2

	deptnum	name	area	mgrnum
size	2	15	1	7
val	10	10	2	10

Optimization Graph (example)



Fragment Reducer Program:

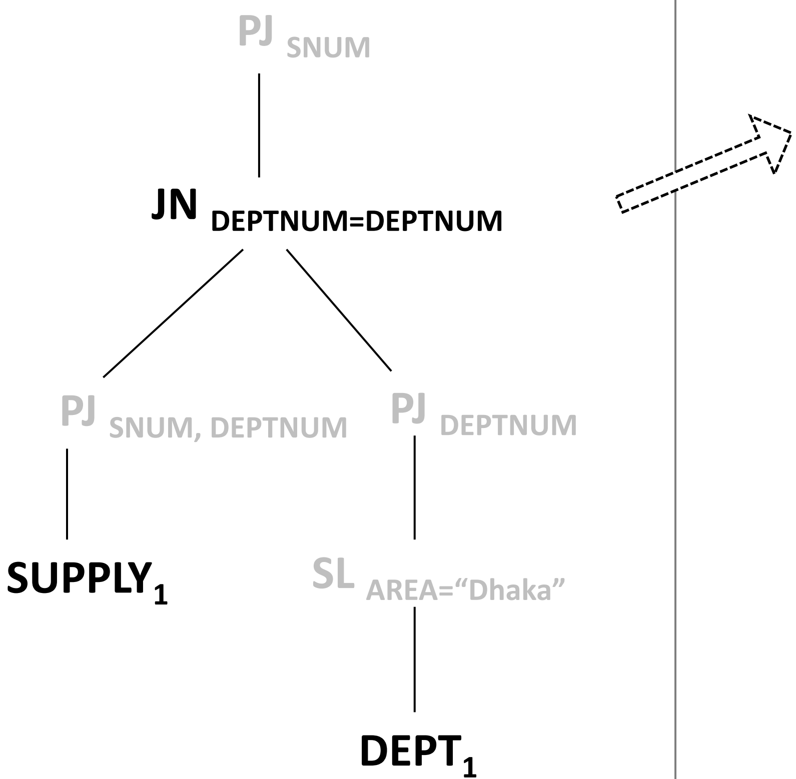
Before binary operation:

- Reducer for $SUPPLY_1$: PJ $_{\text{SNUM, DEPTNUM}}$
- Reducer for $DEPT_1$: PJ $_{\text{DEPTNUM}}$ SL $_{\text{AREA="Dhaka"}}$

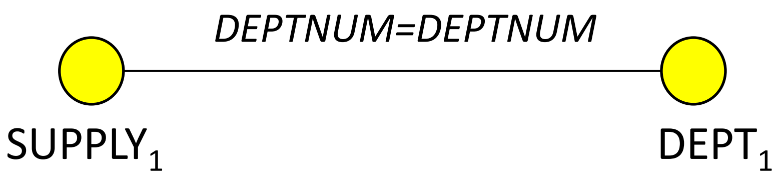
After binary operation:

- Reducer for Result: PJ $_{\text{SNUM}}$

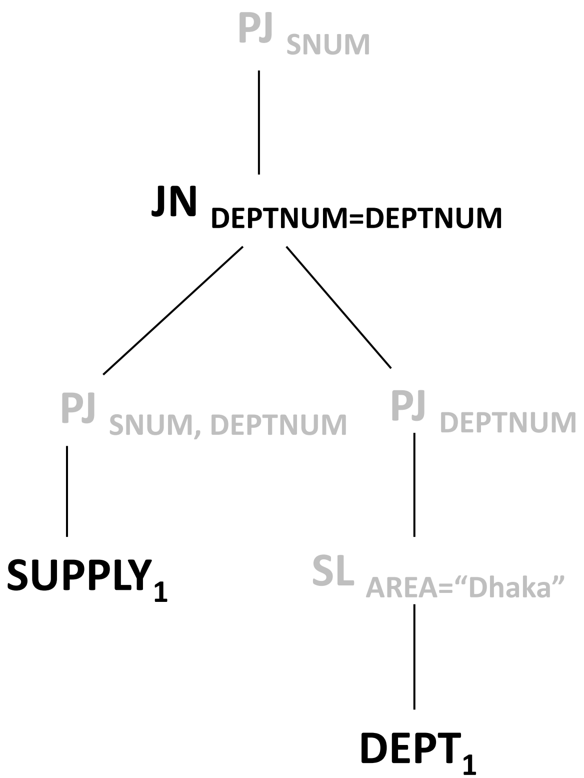
Optimization Graph (example)



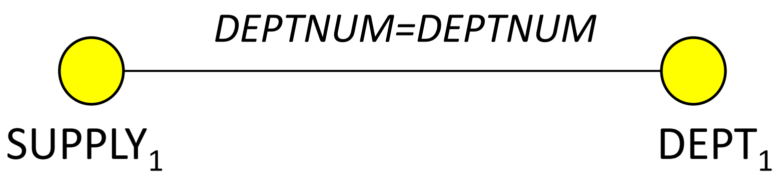
- Optimization Graph:



Optimization Graph (example)



- Optimization Graph:



- Reduced profiles:

	snum	deptnum
size	6	2
val	1800	20

$SUPPLY_1$

	deptnum
size	2
val	10

$DEPT_1$

Additional Reading

- Significance of the *detailed characterization* of the formulas of TC(x) and TD(x).
- Advantages of optimization graph.
- Representing UN’s in optimization graph.
- Assumptions for distributed query optimization.

Practice Problems/ Questions