Principles of Distributed and Parallel Database Systems: Part 2

Total Cost of a Query Execution Plan

Ira A. Fulton Schools of Engineering
Arizona State University

Objectives

Objective

Explain the reliability & fault tolerance concepts in distributed data processing systems

Total Cost

Total cost= CPU cost + I/O cost + communication cost

CPU cost= unit instruction cost $*$ no. of instructions

I/O cost= unit disk I/O cost $*$ no. of disk I/Os

communication cost = message initiation + transmission

Response Time

Response time= CPU time + I/O time + communication time

CPU time= unit instruction time * no. of sequential instructions

I/O time= unit I/O time*no. of sequential I/Os

communication time= unit msg initiation time*no. of sequential msg

+ unit transmission time*no. of sequential bytes

Optimization Statistics

- Primary cost factor:size of intermediate relations
  - Need to estimate their sizes
- Make them precise $\Rightarrow$ more costly to maintain
- Simplifying assumption: uniform distribution of attribute values in a relation

Statistics

- For each relation $R[A_1, A_2, \ldots, A_n]$ fragmented as $R_1, \ldots, R_r$
  - length of each attribute: $length(A_i)$
  - The number of distinct values for each attribute in each fragment: $card(\Pi_{A_i} R_j)$
  - maximum and minimum values in the domain of each attribute: $min(A_i)$, $max(A_i)$
  - the cardinalities of each domain: $card(dom[A_i])$
- The cardinalities of each fragment: $card(R_j)$
  Selectivity factor (SF) of each operation for relations
  - For joins

$$SF_{\bowtie}(R,S) = \frac{card(R \bowtie S)}{card(R) * card(S)}$$

Intermediate Relation Sizes

- Selection
  - $size(R) = card(R) \times length(R)$
  - $card(\sigma_F(R)) = SF_\sigma(F) \times card(R)$

- $S\ F_\sigma(A = value) = ?$
- $S\ F_\sigma(A > value) = ?$
- $S\ F_\sigma(A < value) =$

- length of each attribute: $length(A_i)$
- number of distinct values for each attribute in each fragment: $card(\Pi_{A_i} R_j)$
- maximum and minimum values in the domain of each attribute: $min(A_i)$, $max(A_i)$
- the cardinalities of each domain: $card(dom[A_i])$

Intermediate Relation Sizes

$size(R) = card(R) \times length(R)$
$card(\sigma_F(R)) = SF_\sigma(F) \times card(R)$

where

$$SF_\sigma(A = value) = \frac{1}{card(\prod_A(R))} \qquad SF_\sigma(A > value) = \frac{max(A) - value}{max(A) - min(A)} \qquad SF_\sigma(A < value) = \frac{value - min(A)}{max(A) - min(A)}$$

$$SF_\sigma(p(A_i) \wedge p(A_j)) = SF_\sigma(p(A_i)) \times SF_\sigma(p(A_j))$$

$$SF_\sigma(p(A_i) \vee p(A_j)) = SF_\sigma(p(A_i)) + SF_\sigma(p(A_j)) - (SF_\sigma(p(A_i)) \times SF_\sigma(p(A_j)))$$

Intermediate Relation Sizes

Projection

- $card(\Pi_A(R)) = card(R)$

Cartesian Product

- $card(R \times S) = ?$

Union

- upper bound: $card(R \cup S) = ?$
- lower bound: $card(R \cup S) = ?$

Set Difference

- upper bound: $?$
- lower bound:?

Intermediate Relation Sizes

Projection

- $card(\Pi_A(R)) = card(R)$

Cartesian Product

- $card(R \times S) = card(R) * card(S)$

Union

- upper bound: $card(R \cup S) = card(R) + card(S)$
- lower bound: $card(R \cup S) = max\{card(R), card(S)\}$

Set Difference

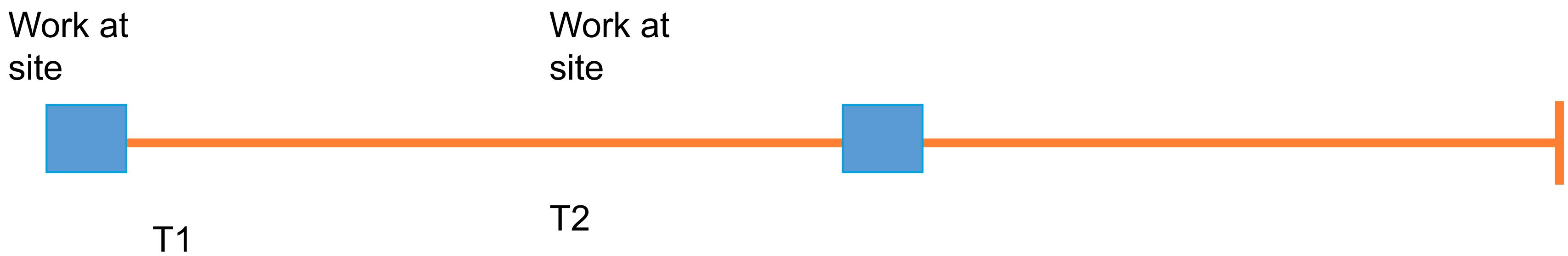
- upper bound: $card(R - S) = card(R)$
- lower bound: 0

Intermediate Relation Size: Join

- **Special case: A is a key of R and B is a foreign key of S**
  - $card(R \bowtie_{A=B} S) = card(S)$

- **More general:**
  - $card(R \bowtie S) = SF_{\bowtie} * card(R) \times card(S)$

IOs May Not be the Best Metric



>>>---------TIME--------->

or $

IOs May Not be the Best Metric

Plan A

100 IOs

Plan B

site 1 50 IOs

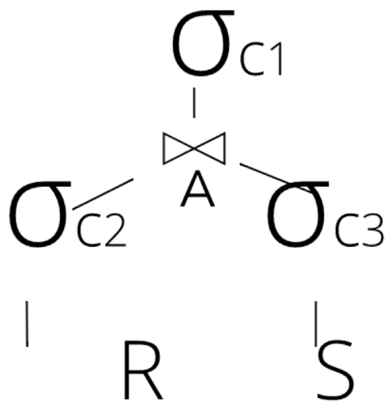
site 2 70 IOs

site 3 50 IOs

Query Separation

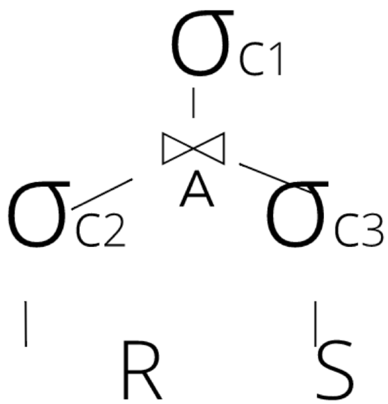
- Separate query into 2 or more steps
- Optimize each step independently

Example: Simple Queries



1. Compute $R' = \Pi A[\sigma c2R]$
   $S' = \Pi A[\sigma c3S]$
2. 2. Compute $J = R' \quad S'$
3. Compute $J = R' \bowtie S'$

Example: Simple Queries



1. Compute $R' = \Pi A[\sigma c2R]$

   $S' = \Pi A[\sigma c3S]$

2. Compute $J = R' \bowtie S'$

3. Compute

$Ans = \sigma c1\{[J \bowtie \sigma c2R]\ \ [J \bowtie \sigma c3S]\}$

Simple Query

- Relations have a single attribute
- Output has a single attribute

$$J \leftarrow R' \bowtie S'$$

Decompose query into

- Local processing
- Simple query (or queries)
- Final processing

Optimize simple query

Philosophy

- Hard part is distributed join
- Do this part with only keys; get rest of data later
- Simpler to optimize simple queries