Query decomposition and data localization

Data Management for Big Data
2018-2019 (spring semester)

Dario Della Monica

These slides are a modified version of the slides provided with the book
Özsu and Valduriez, *Principles of Distributed Database Systems* (3rd Ed.), 2011
The original version of the slides is available at: extras.springer.com

Outline (distributed DB)

- Introduction (Ch. 1) *
- Distributed Database Design (Ch. 3) *
- Distributed Query Processing (Ch. 6-8) *
  - ➡ Overview (Ch. 6) *
  - ➡ **Query decomposition and data localization (Ch. 7)** *
  - ➡ Distributed query optimization (Ch. 8) *
- Distributed Transaction Management (Ch. 10-12) *

* Özsu and Valduriez, *Principles of Distributed Database Systems* (3rd Ed.), 2011

Outline (today)

- Query decomposition and **data localization** (Ch. 7) *
  - ➡ The problem of distributed data localization
  - ➡ A naïve algorithm
  - ➡ Optimization steps (reductions)
    - PHF (selection, join)
    - VF (projection)
    - DHF (selection, join)
    - Hybrid Fragmentation (selection/join + projection)

* Özsu and Valduriez, *Principles of Distributed Database Systems* (3rd Ed.), 2011

Data Localization

Input: Relational algebra expression on global, distributed relations (distributed query)

- Determine which fragments are involved in a query (over global, distributed relations) and transform such a query into an equivalent one over such fragments (localized query)
- Localization uses information about distribution of fragments stored in the fragment schema
- Recall that fragmentation is obtained by several application of rules expressed by relational algebra …
  - ➡ primary horizontal fragmentation: selection σ
  - ➡ derived horizontal fragmentation: semijoin $\ltimes$
  - ➡ vertical fragmentation: projection Π
- … and that reconstruction (reverse fragmentation) rules are also expressed in relational algebra
  - ➡ horizontal fragmentation: union $\cup$
  - ➡ vertical fragmentation: join $\bowtie$

A naïve algorithm to localize distribute queries

- Localization program: relational algebra expression that reconstructs a global relation from its fragments, by reverting the rules employed for fragmentation
- A localized query is obtained from distributed, global query by replacing leaves (global relations) with (the tree of) its corresponding localization program
  - ➡ Leaves of localized queries are fragments
- This approach to obtain a localized query from a distributed one is inefficient and the result can be improved through several optimizations

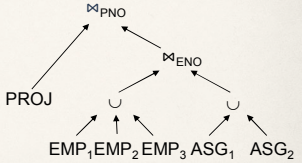
Example

Assume

- EMP is fragmented into EMP_1, EMP_2, EMP_3 as follows:
  - ➡ $EMP_1 = \sigma_{ENO \leq "E3"}(EMP)$
  - ➡ $EMP_2 = \sigma_{"E3" < ENO \leq "E6"}(EMP)$
  - ➡ $EMP_3 = \sigma_{ENO > "E6"}(EMP)$
- ASG fragmented into ASG_1 and ASG_2 as follows:
  - ➡ $ASG_1 = \sigma_{ENO \leq "E3"}(ASG)$
  - ➡ $ASG_2 = \sigma_{ENO > "E3"}(ASG)$

Replace EMP by $(EMP_1 \cup EMP_2 \cup EMP_3)$ and ASG by $(ASG_1 \cup ASG_2)$ in any query

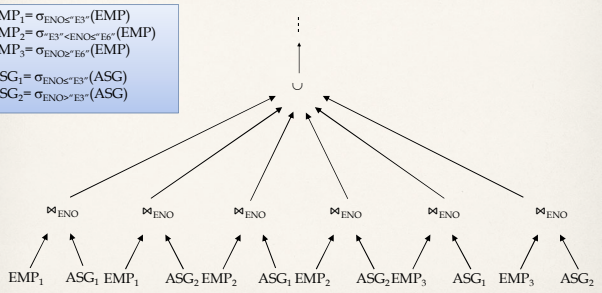
PROJ ⋈ (EMP ⋈ ASG)



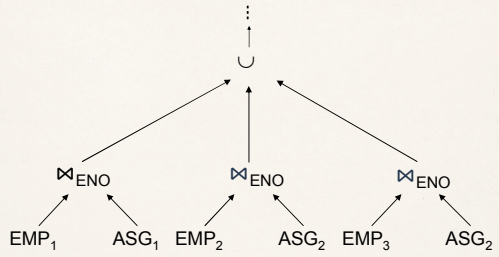
Provides Parallellism

$EMP_1 = \sigma_{ENO \le "E3"}(EMP)$
$EMP_2 = \sigma_{"E3" < ENO \le "E6"}(EMP)$
$EMP_3 = \sigma_{ENO > "E6"}(EMP)$

$ASG_1 = \sigma_{ENO \le "E3"}(ASG)$
$ASG_2 = \sigma_{ENO > "E3"}(ASG)$



Eliminates Unnecessary Work



Identify (pairs of) fragments that can be ignored because they produce empty relations (e.g., when a selection or a join is applied to them)

Reduction for PHF – Selection

- Reduction with selection (ignore a fragment if selection predicate and fragment predicate are contradictory)
  - Relation R and $F_R = \{R_1, R_2, \ldots, R_n\}$ where $R_j = \sigma_{p_j}(R)$

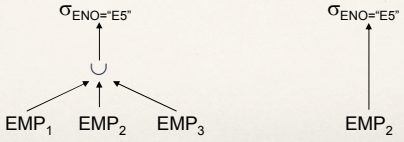
    $\sigma_{p_i}(R_j) = \varnothing$ if $\forall x$ in R: $\neg(p_i(x) \wedge p_j(x))$

  - Example

```
SELECT   *
FROM     EMP
WHERE    ENO="E5"
```

$EMP_1 = \sigma_{ENO \le "E3"}(EMP)$
$EMP_2 = \sigma_{"E3" < ENO \le "E6"}(EMP)$
$EMP_3 = \sigma_{ENO > "E6"}(EMP)$

$ASG_1 = \sigma_{ENO \le "E3"}(ASG)$
$ASG_2 = \sigma_{ENO > "E3"}(ASG)$



Reduction for PHF – Join

- Reduction with join (ignore the join of 2 fragments if their fragment predicates are contradictory over the join attributes)
  - Possible if fragmentation is done on join attribute
  - Distribute join over union

$$R \bowtie S \Leftrightarrow (R_1 \cup R_2) \bowtie (S_1 \cup S_2)$$
$$\Leftrightarrow (R_1 \bowtie S_1) \cup (R_1 \bowtie S_2) \cup (R_2 \bowtie S_1) \cup (R_2 \bowtie S_2)$$

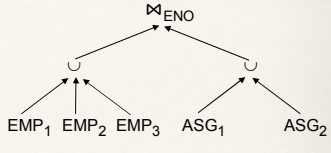
  - Then, join between 2 fragments can be simplified in some cases
    - Given $R_i = \sigma_{p_i}(R)$ and $R_j = \sigma_{p_j}(R)$ [p_i and p_j defined over join attributes]

$R_i \bowtie R_j = \varnothing$ if $\forall x$ in R_i, $\forall y$ in R_j: $\neg(p_i(y) \wedge p_j(x))$

Reduction for PHF – Join (Example)

$EMP_1 = \sigma_{ENO \le "E3"}(EMP)$
$EMP_2 = \sigma_{"E3" < ENO \le "E6"}(EMP)$
$EMP_3 = \sigma_{ENO > "E6"}(EMP)$

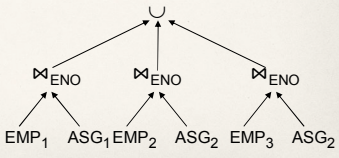
$ASG_1 = \sigma_{ENO \le "E3"}(ASG)$
$ASG_2 = \sigma_{ENO > "E3"}(ASG)$



- Consider the query

```
SELECT   *
FROM     EMP, ASG
WHERE    EMP.ENO=ASG.ENO
```

- Distribute join over unions
- Apply the reduction rule

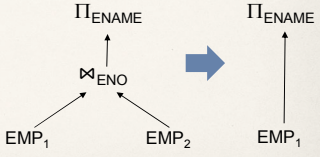


Reduction for VF

- Recall that the localization program consists in joins over key attributes
- Let R be a relation over set of attributes $\{A_1, \ldots, A_n\}$ and R_1 be a fragment of R obtained as $R_1 = \Pi_{A'}(R)$ where $A' \subseteq \{A_1, \ldots, A_n\}$:
  - Reduction for a projection Π_{A_P} over a R_1 is possible when set A_P of projection attributes intersected with set A' of fragmentation attributes is contained in the primary key

Ex.: $EMP_1 = \Pi_{ENO,ENAME}(EMP)$
$EMP_2 = \Pi_{ENO,TITLE}(EMP)$

```
SELECT   ENAME
FROM     EMP
```



Reduction for DHF

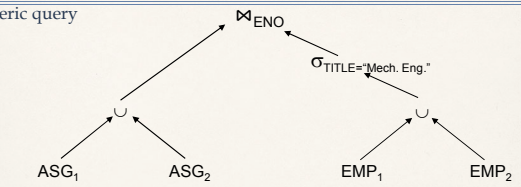
- Similar to the case PHF
- DHF: 2 relations R (member) and S (owner) in association one-to-many
  - R participates with cardinality 1, S participates with cardinality N
  - R can be fragmented following fragmentation on S
  - Fragments that agree on the values of join attributes are placed at the same site
  - Localization program: union
- Rule :
  - Distribute joins over unions
  - Apply the join reduction for horizontal fragmentation
- Example [ASG is member, EMP is owner]

  EMP_1: $\sigma_{TITLE="Programmer"}$ (EMP)
  EMP_2: $\sigma_{TITLE\neq"Programmer"}$ (EMP)
  ASG_1: ASG $\ltimes_{ENO}$ EMP_1
  ASG_2: ASG $\ltimes_{ENO}$ EMP_2
- Query

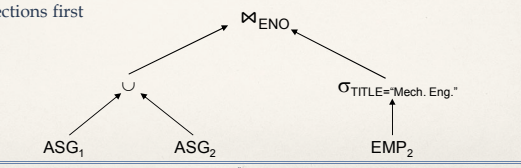
```
SELECT   *
FROM     EMP, ASG
WHERE    ASG.ENO = EMP.ENO
AND      EMP.TITLE = "Mech. Eng."
```


Reduction for DHF

Generic query

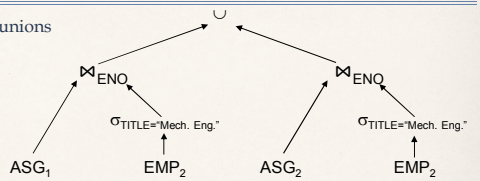


Selections first

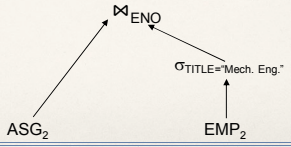


Reduction for DHF

Push join inside unions



Elimination of the empty intermediate relations (left sub-tree)



Reduction for Hybrid Fragmentation

- Combine the rules already specified
  - Remove empty relations generated by contradicting predicates (inside selections or joins) on horizontal fragments
  - Remove useless relations generated by projections on vertical fragments
  - Distribute joins/selections/projections over unions in order to isolate and remove useless operands

Reduction for Hybrid Fragmentation

Example

Consider the following hybrid fragmentation:

$EMP_1 = \sigma_{ENO\leq"E4"}(\Pi_{ENO,ENAME}(EMP))$

$EMP_2 = \sigma_{ENO>"E4"}(\Pi_{ENO,ENAME}(EMP))$

$EMP_3 = \Pi_{ENO,TITLE}(EMP)$

and the query

```
SELECT ENAME
FROM   EMP
WHERE  ENO="E5"
```