Query decomposition and data localization

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) *

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)

Data Localization

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

Output: Relational algebra expression on fragments (localized query)

• Localization uses global information about distribution of fragments (no optimization, no use of quantitative information, e.g., catalog statistics)

• Recall that fragmentation is obtained by several application of rules expressed by relational algebra …
  ➔ primary horizontal fragmentation: selection $\sigma$
  ➔ derived horizontal fragmentation: semijoin $\bowtie$
  ➔ vertical fragmentation: projection $\Pi$

• … 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.
  - During data localization there is a **first optimization phase**
    - we call it **reduction**
    - different from the **proper optimization phase** (finding the “best” strategy for executing the query).
Example

\[ \text{PROJ} \bowtie ( \text{EMP} \bowtie \text{ASG} ) \]
Example

Assume

• EMP is fragmented as follows:
  
  → EMP₁ = σ_{ENO \leq "E3"}(EMP)
  
  → EMP₂ = σ_{"E3" < ENO \leq "E6"}(EMP)
  
  → EMP₃ = σ_{ENO \geq "E6"}(EMP)

• ASG is fragmented as follows:
  
  → ASG₁ = σ_{ENO \leq "E3"}(ASG)
  
  → ASG₂ = σ_{ENO > "E3"}(ASG)

PROJ ⋈ (EMP ⋈ ASG)
Example

Assume

- EMP is fragmented as follows:
  - EMP\(_1\) = \(\sigma_{ENO \leq "E3"}(EMP)\)
  - EMP\(_2\) = \(\sigma_{"E3" < ENO \leq "E6"}(EMP)\)
  - EMP\(_3\) = \(\sigma_{ENO \geq "E6"}(EMP)\)

- ASG is fragmented 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 \(\bowtie\) (EMP \(\bowtie\) ASG)
Example

Assume

• EMP is fragmented as follows:
  
  → EMP\(_1\) = \(\sigma_{\text{ENO} \leq \text{"E3"}}\)(EMP)
  
  → EMP\(_2\) = \(\sigma_{\text{"E3"} < \text{ENO} \leq \text{"E6"}}\)(EMP)
  
  → EMP\(_3\) = \(\sigma_{\text{ENO} \geq \text{"E6"}}\)(EMP)

• ASG is fragmented as follows:
  
  → ASG\(_1\) = \(\sigma_{\text{ENO} \leq \text{"E3"}}\)(ASG)
  
  → ASG\(_2\) = \(\sigma_{\text{ENO} > \text{"E3"}}\)(ASG)

Replace EMP by (EMP\(_1\) \cup EMP\(_2\) \cup EMP\(_3\)) and ASG by (ASG\(_1\) \cup ASG\(_2\)) in any query.
Assume

- **EMP** is fragmented as follows:
  - \( EMP_1 = \sigma_{ENO \leq "E3"}(EMP) \)
  - \( EMP_2 = \sigma_{"E3" < ENO \leq "E6"}(EMP) \)
  - \( EMP_3 = \sigma_{ENO \geq "E6"}(EMP) \)

- **ASG** is fragmented 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.

\[
\text{PROJ} \Join (\text{EMP} \Join \text{ASG}) = \text{PROJ} \Join ((\text{EMP}_1 \cup \text{EMP}_2 \cup \text{EMP}_3) \Join (\text{ASG}_1 \cup \text{ASG}_2))
\]
Provides Parallellism

(EMP₁ ∪ EMP₂ ∪ EMP₃) ⋈ (ASG₁ ∪ ASG₂)
Provides Parallellism

\[(\text{EMP}_1 \cup \text{EMP}_2 \cup \text{EMP}_3) \bowtie (\text{ASG}_1 \cup \text{ASG}_2) = (\text{EMP}_1 \bowtie \text{ASG}_1) \cup (\text{EMP}_1 \bowtie \text{ASG}_2) \cup (\text{EMP}_2 \bowtie \text{ASG}_1) \cup (\text{EMP}_2 \bowtie \text{ASG}_2) \cup (\text{EMP}_3 \bowtie \text{ASG}_1) \cup (\text{EMP}_3 \bowtie \text{ASG}_2)\]
Provides Parallelism

(EMP₁ ⋈ EMP₂ ⋈ EMP₃) ⋈ (ASG₁ ⋈ ASG₂)

= 

(EMP₁ ⋈ ASG₁) ⋈ (EMP₁ ⋈ ASG₂) ⋈ (EMP₂ ⋈ ASG₁) ⋈ (EMP₂ ⋈ ASG₂) ⋈ (EMP₃ ⋈ ASG₁) ⋈ (EMP₃ ⋈ ASG₂)
Provides Parallellism

\[\begin{align*}
\text{EMP}_1 &= \sigma_{\text{ENO} \leq \text{"E3"}}(\text{EMP}) \\
\text{EMP}_2 &= \sigma_{\text{"E3"} < \text{ENO} \leq \text{"E6"}}(\text{EMP}) \\
\text{EMP}_3 &= \sigma_{\text{ENO} \geq \text{"E6"}}(\text{EMP}) \\
\text{ASG}_1 &= \sigma_{\text{ENO} \leq \text{"E3"}}(\text{ASG}) \\
\text{ASG}_2 &= \sigma_{\text{ENO} > \text{"E3"}}(\text{ASG})
\end{align*}\]
Provides Parallellism

EMP₁ = σENO≤"E3"(EMP)
EMP₂ = σ"E3"<ENO≤"E6"(EMP)
EMP₃ = σENO≥"E6"(EMP)

ASG₁ = σENO≤"E3"(ASG)
ASG₂ = σENO>"E3"(ASG)

(EMP₁ ∪ EMP₂ ∪ EMP₃) ⊙ (ASG₁ ∪ ASG₂) =
(EMP₁ ⊙ ASG₁) ∪ (EMP₁ ⊙ ASG₂) ∪
(EMP₂ ⊙ ASG₁) ∪ (EMP₂ ⊙ ASG₂) ∪
(EMP₃ ⊙ ASG₁) ∪ (EMP₃ ⊙ ASG₂)
Provides Parallelism

\[
\begin{align*}
\text{EMP}_1 &= \sigma_{\text{ENO} \leq \text{"E3"}}(\text{EMP}) \\
\text{EMP}_2 &= \sigma_{\text{"E3"} < \text{ENO} \leq \text{"E6"}}(\text{EMP}) \\
\text{EMP}_3 &= \sigma_{\text{ENO} \geq \text{"E6"}}(\text{EMP}) \\
\text{ASG}_1 &= \sigma_{\text{ENO} \leq \text{"E3"}}(\text{ASG}) \\
\text{ASG}_2 &= \sigma_{\text{ENO} > \text{"E3"}}(\text{ASG})
\end{align*}
\]

\[ (\text{EMP}_1 \cup \text{EMP}_2 \cup \text{EMP}_3) \bowtie (\text{ASG}_1 \cup \text{ASG}_2) = \]
\[ (\text{EMP}_1 \bowtie \text{ASG}_1) \cup (\text{EMP}_1 \bowtie \text{ASG}_2) \cup (\text{EMP}_2 \bowtie \text{ASG}_1) \cup (\text{EMP}_2 \bowtie \text{ASG}_2) \cup (\text{EMP}_3 \bowtie \text{ASG}_1) \cup (\text{EMP}_3 \bowtie \text{ASG}_2) \]
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 of a selection over a relation fragmented with PHF (ignore a fragment if selection predicate and fragment predicate are contradictory)
  
  - Consider $\sigma_p(R)$
  
  - Horizontal fragmentation on $R$: $F_R = \{R_1, R_2, \ldots, R_w\}$, where $R_j = \sigma_{p_j}(R)$
  
  - $\sigma_p(R_j) = \emptyset$ if $\forall x$ in $R$: $\neg(p(x) \land p_j(x))$ i.e., $p$ and $p_j$ are contradictory
Reduction for PHF – Selection (Example)

- Reduction of a selection over a relation fragmented with PHF (ignore a fragment if selection predicate and fragment predicate are contradictory)

→ Example

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

**Distributed query**

```
σ_{ENO="E5"}(EMP) ∪ σ_{ENO="E5"}(EMP_1) ∪ σ_{ENO="E5"}(EMP_2) ∪ σ_{ENO="E5"}(EMP_3)
```

**Localized query**

```
EMP_1 = σ_{ENO≤"E3"}(EMP) 
EMP_2 = σ_{E3<ENO≤"E6"}(EMP) 
EMP_3 = σ_{ENO≥"E6"}(EMP) 
```

**Reduced local query**

```
ASG_1 = σ_{ENO≤"E3"}(ASG) 
ASG_2 = σ_{ENO>"E3"}(ASG)
```
Reduction for PHF – Join

• Reduction of a join over relations fragmented with PHF (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

    \[ R \bowtie S \iff (R_1 \cup R_2) \bowtie (S_1 \cup S_2) \]
    \[ \iff (R_1 \bowtie S_1) \cup (R_1 \bowtie S_2) \cup (R_2 \bowtie S_1) \cup (R_2 \bowtie S_2) \]

  → Then, join between 2 fragments can be simplified in some cases

  ✦ Given \( R_i = \sigma_{p_i}(R) \) and \( S_j = \sigma_{p_j}(S) \) \([p_i \text{ and } p_j \text{ defined over join attributes}]\)

    \[ R_i \bowtie S_j = \emptyset \text{ if } \forall x \in R \cup S: \neg (p_i(x) \land p_j(x)) \]
    \[ \text{[there is a mistake in the textbook]} \]
    \[ \text{i.e., } p_i \text{ and } p_j \text{ are contradictory} \]
Reduction for PHF – Join (Example)

- Consider the query

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

- Distribute join over unions

- Apply the reduction rule

\[
\begin{align*}
\text{EMP}_1 &= \sigma_{\text{ENO} \leq \text{"E3"}}(\text{EMP}) \\
\text{EMP}_2 &= \sigma_{\text{"E3"} < \text{ENO} \leq \text{"E6"}}(\text{EMP}) \\
\text{EMP}_3 &= \sigma_{\text{ENO} \geq \text{"E6"}}(\text{EMP}) \\
\text{ASG}_1 &= \sigma_{\text{ENO} \leq \text{"E3"}}(\text{ASG}) \\
\text{ASG}_2 &= \sigma_{\text{ENO} > \text{"E3"}}(\text{ASG})
\end{align*}
\]
Reduction for PHF – Join (Example)

- Consider the query
  
  ```
  SELECT * 
  FROM EMP, ASG 
  WHERE EMP.ENO = ASG.ENO 
  ```

- Distribute join over unions
- Apply the reduction rule

\[
\begin{align*}
\text{EMP}_1 &= \sigma_{\text{ENO} \leq \text{"E3"}}(\text{EMP}) \\
\text{EMP}_2 &= \sigma_{\text{"E3"} < \text{ENO} \leq \text{"E6"}}(\text{EMP}) \\
\text{EMP}_3 &= \sigma_{\text{ENO} \geq \text{"E6"}}(\text{EMP}) \\
\text{ASG}_1 &= \sigma_{\text{ENO} \leq \text{"E3"}}(\text{ASG}) \\
\text{ASG}_2 &= \sigma_{\text{ENO} > \text{"E3"}}(\text{ASG})
\end{align*}
\]

Not always convenient
Reduction for VF

- Reduction of a projection over a relation fragmented with VF (ignore the fragment for which the set of projection attributes intersected with set of fragmentation attributes is contained in the primary key)
- Recall that the localization program consists in joins over key attributes
- Let $R_1$ be a fragment of $R$ obtained as $R_1 = \Pi_{A'}(R)$ where $A' \subseteq \text{attr}(R)$:
  - Reduction of a projection $\Pi_{A''}$ over $R_1$ is possible when $A'' \cap A' \subseteq \text{key}(R)$

Ex.: $\text{EMP}_1 = \Pi_{\text{ENO},\text{ENAME}}(\text{EMP})$
$\text{EMP}_2 = \Pi_{\text{ENO},\text{TITLE}}(\text{EMP})$

```
SELECT ENAME
FROM EMP
```
Reduction for DHF

- Similar to the case PHF
- DHF: 2 relations $S$ (owner) and $R$ (member) in association one-to-many
  - $S$ participates with cardinality $N$, $R$ participates with cardinality 1
  - Fragmentation propagate from $S$ to $R$
  - Localization program: union
  - Fragments that agree on the values of join attributes are placed at the same site
- Rule:
  - Distribute joins over unions
  - Apply the join reduction for horizontal fragmentation
Reduction for DHF – Example

• Example  
  [EMP is owner, ASG is member]

  EMP₁: \( \sigma_{\text{TITLE="Programmer"}} \) (EMP)
  EMP₂: \( \sigma_{\text{TITLE \neq "Programmer"}} \) (EMP)
  ASG₁: ASG \( \bowtie \) ENO EMP₁
  ASG₂: ASG \( \bowtie \) ENO EMP₂

• Query

  SELECT *
  FROM EMP, ASG
  WHERE ASG.ENO = EMP.ENO
Reduction for DHF – Example

- **Example**  [EMP is owner, ASG is member]
  
  \[
  \begin{align*}
  \text{EMP}_1: & \; \sigma_{\text{TITLE}=\text{"Programmer"}} (\text{EMP}) \\
  \text{EMP}_2: & \; \sigma_{\text{TITLE} \neq \text{"Programmer"}} (\text{EMP}) \\
  \text{ASG}_1: & \; \text{ASG} \bowtie \text{ENO} \; \text{EMP}_1 \\
  \text{ASG}_2: & \; \text{ASG} \bowtie \text{ENO} \; \text{EMP}_2
  \end{align*}
  \]

- **Query**
  
  \[
  \text{SELECT} \; * \\
  \text{FROM} \; \text{EMP, ASG} \\
  \text{WHERE} \; \text{ASG.ENO} = \text{EMP.ENO}
  \]

Always convenient
- the number of joins is always equal to the number of fragments
- all joins can be performed in parallel (are disjoint)
Complex reduction for PHF and DHF

1. Generic query

\[ \text{ASG}_1 \bowtie \text{ASG}_2 \bowtie \text{EMP}_1 \bowtie \text{EMP}_2 \]

\[ \sigma_{\text{TITLE}="\text{Mech. Eng."}} \]

\[ \cup \]

\[ \cup \]
Complex reduction for PHF and DHF

1. Generic query

2. Reduction of selection over a relation fragmented with HF
Complex reduction for PHF and DHF

1. Generic query

2. Reduction of selection over a relation fragmented with HF
Complex reduction for PHF and DHF

1. Generic query

2. Reduction of selection over a relation fragmented with HF

3. Reduction of join over a relation fragmented with DHF
Complex reduction for PHF and DHF

1. Generic query

2. Reduction of selection over a relation fragmented with HF

3. Reduction of join over a relation fragmented with DHF
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:

\[
\begin{align*}
\text{EMP}_1 &= \sigma_{\text{ENO} \leq "E4"}(\Pi_{\text{ENO,ENAME}}(\text{EMP})) \\
\text{EMP}_2 &= \sigma_{\text{ENO} > "E4"}(\Pi_{\text{ENO,ENAME}}(\text{EMP})) \\
\text{EMP}_3 &= \Pi_{\text{ENO,TITLE}}(\text{EMP})
\end{align*}
\]

Thus, the localization program for EMP is:

\[
\text{EMP} = (\text{EMP}_1 \cup \text{EMP}_2) \bowtie \text{EMP}_3
\]

Consider also the query:

\[
\begin{align*}
\text{SELECT} & \quad \text{ENAME} \\
\text{FROM} & \quad \text{EMP} \\
\text{WHERE} & \quad \text{ENO} = "E5"
\end{align*}
\]