Optimización

Bibliografía: Fundamentos de bases de datos
Korth , Silberschatz

Conceptos básicos

Dada una **consulta**, existen **diferentes formas** (**estrategias**) de calcular la **respuesta**.

Los **usuarios** generalmente **no** escriben la consulta con la **estrategia más eficiente**.

El **sistema** debe **transformar** la consulta del usuario **en una equivalente más eficiente**.

Esto se llama **optimización de consultas**.

Conceptos básicos

El **optimizador** es un ***programa***:

- **capaz** de **considerar** cientos de **estrategias diferentes** para una solicitud,

- para **elegir** una **estrategia eficiente** para evaluar una expresión relacional dada.

Observaciones

- En general se habla de **optimización** de **consultas**, pero podría ser parte de una **actualización**.

- **No siempre** la estrategia elegida es la **óptima**
  - pero es **mejor** que la **no optimizada**.

- El **costo** de procesar una consulta está normalmente dominado por el **acceso al disco**.

Ejemplo para mostrar la necesidad de la optimización

Obtener los nombres de los proveedores que suministran la parte P2.

Una posible formulación en SQL sería:

```
SELECT DISTINCT SNOMBRE
FROM S, SP
WHERE S.S# = SP.S#  AND  SP.P# = 'P2';
```

Ejemplo para mostrar la necesidad de la optimización

```
SELECT DISTINCT SNOMBRE
FROM S, SP
WHERE S.S# = SP.S#  AND  SP.P# = 'P2';
```

Supongamos que:

- la tabla S contiene 100 proveedores
- la tabla SP contiene 10.000 envíos,
- de los cuales sólo 50 envíos son de P2.

Si el sistema la ejecutara sin optimizar, la secuencia sería…

- Calcular el **producto cartesiano** de las relaciones **S y SP**. (S x SP)
  - Esto es **leer 100 veces c/u** de las **10.000 tuplas de envíos** = **1.000.000** tuplas leídas.
  - El **producto** contendrá 1.000.000 tuplas, lo cual **no entra en memoria principal**,
  - por lo cual habrá que **grabarlo en disco** e implica **1.000.000 de grabaciones de tuplas**.

Si el sistema la ejecutara sin optimizar, la secuencia sería…

- **Restringir** el resultado del paso 1 **según el WHERE**.
  - **Leer otra vez 1.000.000** de tuplas,
  - esto produce una relación formada por **50 tuplas**, que sí cabe en memoria principal.

Si el sistema la ejecutara sin optimizar, la secuencia sería…

- **Proyectar** el resultado del paso 2 **sobre SNOMBRE**.
  - esto produce un **máximo de 50 tuplas**.

Si el sistema la ejecutara sin optimizar, la secuencia sería…

- Calcular el **producto cartesiano** de las relaciones **S y SP**. (S x SP)
- **Restringir** el resultado del paso 1 **según el WHERE**.
- **Proyectar** el resultado del paso 2 **sobre SNOMBRE**.

Esto produce 3.000.000 accesos a disco.

Procedimiento equivalente pero más eficiente…

- **Restringir** la relación **SP** a las tuplas de la **parte P2**.
  - Esto es **leer 10.000 tuplas**,
  - pero **produce** una relación formada por **50**
  - que pueden mantenerse en memoria principal.

Procedimiento equivalente pero más eficiente…

- **Reunir** el resultado del paso 1 con la relación **S según S#.**
  - Esto es **leer sólo 100 tuplas**.
  - El **resultado** contiene **50 tuplas** (siguen en memoria principal)

Procedimiento equivalente
pero más eficiente…

- **Proyectar** el resultado del paso 2 **sobre SNOMBRE**.
  - esto produce un **máximo de 50 tuplas**.

Esto produce 10.100 accesos a disco.

Observaciones

Si se adopta el **"número de operaciones de E/S de tuplas"** como **medida de desempeño**,

- el **segundo** de estos procedimientos es **unas 300 veces mejor que el primero**.
- el primero implica 3.000.000 operaciones de E/S de tuplas
- el segundo implica sólo 10.100

Observaciones

3.000.000 / 10.100 ≈ 300

con solo **cambiar el orden de la reunión y selección**.

Observaciones

- Si tabla **SP** estuviera **indizada** según P#
  - El **número de tuplas leídas** en el paso 1 **se reduce** de 10.000 a sólo **50**.
  - En el segundo paso continúa leyendo **100 tuplas**.

3.000.000 / 150 ≈ 20.000

➔ el procedimiento sería **20.000 veces mejor** que el original.

El proceso de optimización: un panorama general

1. Traducir la consulta a una representación interna

- Se utiliza el formalismo del **álgebra relacional**.
- **Cada expresión** del AR representa una **secuencia de operaciones**.
- Esta secuencia se puede representar como un **árbol de consulta**.

La representación algebraica de la consulta podría ser:

$$\pi_{\text{SNOMBRE}} (\sigma_{\text{P = 'P2'}} (\text{S} \mid \text{x} \mid \text{SP}))$$

El proceso de optimización: un panorama general

2. Convertir a una forma equivalente

- Toda consulta puede escribirse de varias formas equivalentes.
- La consulta del ejemplo tiene 8 formas equivalentes
- Se trata de encontrar una representación más eficiente que la anterior.

El proceso de optimización: un panorama general

Criterios:

→ **Efectuar las selecciones antes que las reuniones:**

- **reduce** el tamaño de **entrada a la reunión** (cantidad de datos a revisar)
- **reduce** el tamaño de **salida de la reunión**,
  - ➔ podría ser la diferencia entre **conservar** esa salida en **memoria ppal** o **vaciarla** en **disco**

Ejemplos

$\sigma_{\text{restricción-sobre-B}}$ **(A |X| B) =**

Ejemplos

$\sigma_{\text{restricción-sobre-B}}$ **(A |X| B) =**

(A |x| $\sigma_{\text{restricción-sobre-B}}$ **(B))**

Ejemplos

$\sigma_{\text{restricción-sobre-A AND restricción-sobre-B}}$ **(A |x| B)=**

Ejemplos

$\sigma_{\text{restricción-sobre-A AND restricción-sobre-B}}$ **(A |x| B)=**

($\sigma_{\text{restricción-sobre-A}}$ **(A) |X|** $\sigma_{\text{restricción-sobre-B}}$ **(B))**

→ **CONVERTIR CONDICIÓN DE RESTRICCIÓN A UNA CONDICIÓN EQUIVALENTE EN *FORMA NORMAL CONJUNTIVA***

Forma Normal Conjuntiva (FNC) es una condición formada por un conjunto de restricciones enlazadas mediante el operador lógico **AND** donde **cada restricción** se compone a su vez de un conjunto de comparaciones enlazadas sólo por el operador **OR**

Ejemplo: p, q y r son condiciones de restricción:

$\sigma_{\text{p OR (q AND r)}}$ **(r) =** $\sigma_{\text{(p OR q) AND (p OR r)}}$ **(r)**

Ventajas de la *FNC*:

La evaluación de condición resulta **verdadera** sólo si el resultado de evaluar **cada conjunción es verdadero**

lo que es equivalente: Resulta **falsa** si el resultado de evaluar **cualquiera es falso**

➔ Si el resultado de evaluar **una** de las condiciones es **falso**, **no** hace falta **evaluar** el **resto**

El **optimizador decide** cuál evaluar

→una secuencia de selecciones se puede combinar en una sola selección:

$\sigma_{\text{restricción-2}}(\sigma_{\text{restricción-1}}(A)) =$

→una secuencia de selecciones se puede combinar en una sola selección:

$\sigma_{\text{restricción-2}}(\sigma_{\text{restricción-1}}(A)) =$

$\sigma_{\text{restricción-2 AND restricción-1}}(A)$

$\pi_{\text{proyección-2}}(\pi_{\text{proyección-1}}(A)) =$

→ en una secuencia de proyecciones, se puede hacer caso omiso de todas con excepción de la última

$$\pi_{\text{proyección-2}} (\pi_{\text{proyección-1}} (A)) = \pi_{\text{proyección-2}} (A)$$

Ejemplo:

"Obtener una lista de las ciudades en la vista V", donde V = proyección de la relación S sobre S# y CIUDAD equivale a proyectar directamente S sobre CIUDAD

→ una selección de una proyección es equivalente a una proyección de una selección

$$\sigma_{\text{restricción}} (\pi_{\text{proyección}} (A)) =$$

$$\pi_{\text{proyección}} (\sigma_{\text{restricción}} (A))$$

→ considerar restricciones de integridad

el sistema sabe que SP.P# es una **clave ajena** que **concuerda** con la **clave primaria** P.P#:

$$\pi_{S\#} (SP \ |x| \ P) = \pi_{S\#} (SP)$$

- (SP JOIN P)[S#] = representa los números de **proveedores** que **suministran** por lo menos **una parte**
- se puede **simplificar** a sólo SP [S#] con lo cual se **elimina** una **reunión**.

→ otras equivalencias

$$\sigma_{\text{restric1}} \ \text{AND}_{\text{resctric2}} (A) = \sigma_{\text{restric1}} (\sigma_{\text{restric2}} (A))$$

$$\sigma_{\text{restric1}} (\sigma_{\text{restric2}} (A)) = \sigma_{\text{restric2}} (\sigma_{\text{restric1}} (A))$$

$$\sigma_{\text{restricción}} (A \cup B) = \sigma_{\text{restricción}} (A) \cup \sigma_{\text{restricción}} (B)$$

Observaciones

- Realizar las operaciones de **selección** tan pronto como sea posible
- Realizar pronto las **proyecciones**
  - A partir de la expresión original, el **optimizador** aplica en **forma repetida sus reglas de transformación** hasta llegar por fin **a "la mejor" versión**

3. Estimación de costos de las consultas

→ Luego de convertir la consulta, el optimizador deberá evaluarla, considerando:

- la distribución de los datos almacenados
- la existencia de índices u otras rutas de acceso,
- el agrupamiento físico de los registros, etc.

Estimación de costos de las consultas

→ Existen distintos procedimientos de bajo nivel para realizar las operaciones de proyección, reunión y restricción.

→ Ejemplo: en el caso de selección algunos de los procedimientos consideran:

- si el campo de restricción esta indizado,
- si la condición de igualdad es según una clave candidata,
- si los datos no están indizados pero están agrupados físicamente

Estimación de costos de las consultas

→ Cada procedimiento tiene un costo.

→ El optimizador los elegirá valiéndose de la información en el catálogo:

- nro. de tuplas de cada relación,
- nro. de valores distintos de un atributo A en r ,
- tamaño del registro de una relación
- nro. de valores distintos de los datos en cada índice,
- nro. de páginas ocupadas por cada relación, etc.

Estimación de costos de las consultas

- Esta información está en el catálogo
- Permite estimar el coste

- La actualización del catálogo depende del comando

UPDATE STATISTICS

4. Generar planes de consulta y elegir el más económico

- Construye un conjunto de planes de consulta candidatos
- Elige el mejor, es decir el más económico.
- El costo en esencia es una estimación de las operaciones de E/S de disco requeridas.

Conclusiones

tiempo(optimización)

+

tiempo(consulta optimizada)

< tiempo (consulta sin optimizar)

Se puede reducir el tiempo(optimización)

- determinando una estrategia que es probable que tenga un coste bajo
- analizando otras pocas
- descartando otras sin completar el cálculo de coste