

TEMA 3.

INTRODUCCIÓN AL PROCESAMIENTO DE CONSULTAS DISTRIBUIDAS

Material para revisar en clase.

3.1. EL PROBLEMA DEL PROCESAMIENTO DE CONSULTAS.

- En BDD el procesamiento de consultas es mucho más complicado por el número de parámetros que afectan su rendimiento.
- El uso de SQL oculta los detalles en cuanto a que el usuario no especifica explícitamente el procedimiento para ejecutar la consulta.
- El procesamiento de una consulta se realiza a través del llamado “*procesador de consultas*” liberando entre otras cosas al programador de su optimización.

3.1.1. *Objetivo del procesador distribuido de consultas.*

- El rol principal del procesador distribuido es mapear o transformar una sentencia global en un conjunto de consultas en términos de fragmentos en cada sitio.
- Para lograr este mapeo se realizan las siguientes acciones:
 - Transformar la consulta global en términos de álgebra relacional.
 - Construir una serie de sentencias que serán ejecutadas en cada nodo para recuperar los datos: *Proceso de localización*.
 - Incorporar operadores de comunicación, por ejemplo, operadores de transmisión de datos, intercambio de datos entre nodos, en general operaciones que implican la comunicación entre nodos.
 - Optimizar la consulta considerando el uso de 3 principales recursos:
 - costo de procesamiento,
 - accesos a disco,
 - y en especial costos de comunicación e intercambio de datos entre sitios.
- La transformación de la consulta en términos de álgebra relacional puede generar *múltiples soluciones equivalentes* que crecen exponencialmente al aumentar el número de relaciones involucradas en la consulta.
- La elección de la estrategia más óptima se vuelve muy costosa por lo que se prefiere seleccionar no la mejor, pero sí una solución cercana.
- En BDD el universo de posibles soluciones se incrementa por la naturaleza de distribución de los datos.

Ejemplo 1:

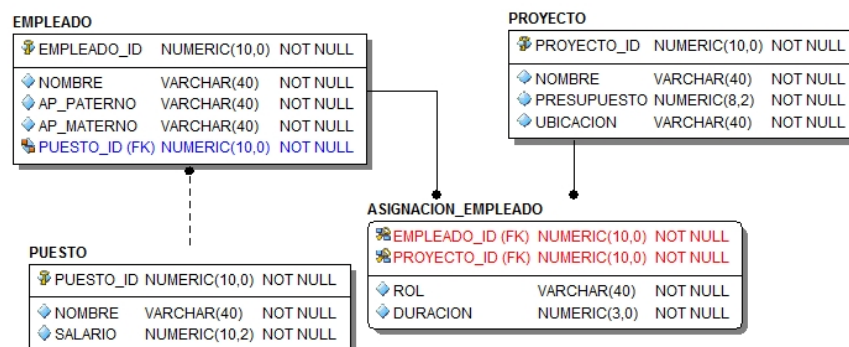
Mostrar el nombre de los empleados que tienen una duración de 37 meses en sus proyectos.

- Generar una consulta SQL global que obtendrá los datos solicitados.
- Realizar 2 transformaciones de la consulta anterior en álgebra relacional equivalentes.

- En BDD se deben incorporar operadores para intercambiar datos entre sitios.
- El procesador de consultas distribuido debe seleccionar al sitio más adecuado para procesar la consulta.

Ejemplo 2:

Suponer que la tabla E y AE fueron fragmentadas con base a las siguientes reglas:



- Empleados cuyo apellido paterno inicia con las letras [A-M]
- Empleados cuyo apellido materno inicia con las letras [N-Z]
- Los datos de AE se distribuyen con base a la fragmentación de E
- Se cuentan con 4 Sitios. En S1, S2 se almacenan los datos de la tabla E, y en S3 y S4 se almacenan los datos de AE

La consulta se lanza en un quinto sitio donde no existen fragmentos.

- Esquema de fragmentación:

$$EMP_1 = \sigma_{\text{substr}(\text{ap-paterno},1,1) \geq 'A' \text{ and } \text{substr}(\text{ap-paterno},1,1) \leq 'M'}(EMP)$$

$$EMP_2 = \sigma_{\text{substr}(\text{ap-paterno},1,1) \geq 'N' \text{ and } \text{substr}(\text{ap-paterno},1,1) \leq 'Z'}(EMP)$$

$$AE_1 = AE \bowtie_{\text{emp-id}} EMP_1$$

$$AE_2 = AE \bowtie_{\text{emp-id}} EMP_2$$

- A. Estrategia de procesamiento:

Describir los pasos que realizará el DDBMS para ejecutar la consulta (ver figura).

- B. Calcular el costo de la estrategia anterior considerando los siguientes datos:

- Las tuplas están distribuidas de forma uniforme en los fragmentos con respecto al predicado $\text{duracion} = 37$; es decir:

$$\text{Card}(\sigma_{\text{duracion}=37}(AE)) = 20$$

- $\text{Card}(AE) = 1000$
- $\text{Card}(EMP) = 400$
- Costo de acceder a una tupla = 1
- Costo de transmitir una tupla a otro sitio = 10
- Existe un índice en el campo duración en AE.

El costo total será:

$$c_1 = \text{Acceder tuplas en } S'_3 = 10 * \text{costo de acceso} = 10 * 1 = \mathbf{10}$$

$$c_2 = \text{Transmitir tuplas hacia } S_2 = 10 * \text{costo transmision} = 10 * 10 = \mathbf{100}$$

$$c_3 = \text{Acceder tuplas en } S_2 \text{ para ejecutar el join} = 10 * 2 = \mathbf{20}.$$

Se multiplica por 2, uno para E y otro para S'_3 , se hace uso del índice. Por ello solo se tiene un costo de 20.

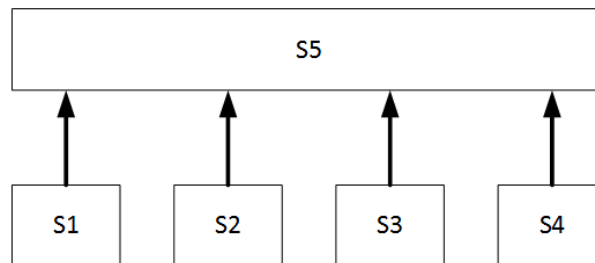
$$c_4 = \text{Transferir tuplas a } S_5 = 10 * \text{costo transmision} = 10 * 10 = \mathbf{100}$$

$$c_t = 2 * (c_1 + c_2 + c_3 + c_4) = 2 * (10 + 100 + 20 + 100) = \mathbf{460}, \text{ Se multiplica por 2 ya que el mismo costo ocurre con } S_2 \rightarrow S_3 \rightarrow S_5$$

Ejercicio 3:

Empleando los mismos datos, calcular el costo total considerando la siguiente estrategia de procesamiento:

- Todos los sitios procesan y envían resultados a S5



Respuesta:

$c_1 = \text{transferir } E_1 \text{ y } E_2 \text{ a } S_5 = 400 * \text{costo transmisión} = 400 * 10 = \mathbf{4,000}$

$c_2 = \text{transferir } AE_1 \text{ y } AE_2 \text{ a } S_5 = 1000 * \text{costo transmisión} = 1000 * 10 = \mathbf{10,000}$

$c_3 = \text{ejecutar } R_1 = \sigma_{\text{duracion}=37}(AE_1 \cup AE_2) = 1000 \text{ tuplas} * \text{costo de acceso} = \mathbf{1000}$

aquí no existe índice, por lo tanto se leen los 1000 registros.

$c_4 = \text{ejecutar } R_1 \bowtie_{\text{emp-id}} EMP = 20 \text{ tuplas de } R_1 * 400 \text{ tuplas de } EMP * \text{costo de acceso} = 20 * 400 * 1 = \mathbf{8000}$,
por cada tupla de R_1 se leen las 400 tuplas de EMP (Nested Join)

$c_t = c_1 + c_2 + c_3 + c_4 = 4,000 + 10,000 + 1,000 + 8,000 = \mathbf{23,000}$

3.2. ELEMENTOS QUE INFLUYEN EN LA OPTIMIZACIÓN DE CONSULTAS

- En general, el proceso de optimización trata de minimizar los costos de la siguiente función:

$$c = \text{costo } I/O + \text{costo procesador} + \text{costo comunicación}$$

Las siguientes variables se toman en cuenta para minimizar el costo de la fórmula anterior:

- Cardinalidad de los fragmentos
- Estadísticas.
- Uso de semi Joins.
- Cantidad de datos a transmitir
- Complejidad de las operaciones del álgebra relacional a emplear:

Operación	Complejidad (Completar)
Selección	
Proyección	
Proyección	
Group by	
Join	
Semi-Join	
Set Operators (Union, Union all, intersect, minus)	
Producto cartesiano	

Donde:

- n representa la cardinalidad de la relación.
- $O(n \log(n))$ en general aplica para operadores binarios donde se asume que se requiere hacer comparación de los valores entre tuplas en donde se requiere un ordenamiento previo de una de las relaciones para poder comparar.
- Empleando técnicas con tablas Hash la complejidad anterior se puede reducir a $O(n)$

El tipo de optimizador también influye en el proceso de optimización.

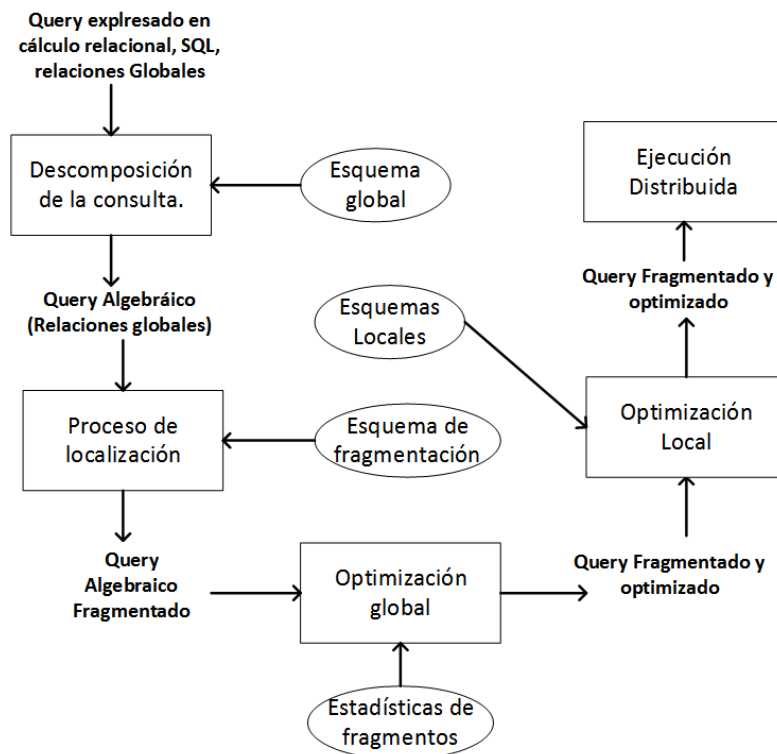
3.2.1. Tipos de optimizadores basados en sus características

- Optimizador basado en técnicas de búsqueda exhaustiva.**
 - Genera las N posibles combinaciones que pueden existir para ejecutar una consulta.
 - Se puede tener una complejidad $O(n^N)$ donde N es el número de tablas que participan en un Join, esto puede representar un problema de tiempos y procesamiento.
 - La selección del resultado se basa en el cálculo de costos.
 - La solución obtenida es óptima
- Optimizador basado en heurísticas**
 - La solución que genera no necesariamente es la mejor, pero si cercana a la óptima.
 - Aplica Heurísticas como:
 - Realiza agrupación de operaciones similares

- Ejecuta operaciones con bajo costo al inicio (selección, proyección, etc.)
 - Reemplaza el uso de Joins por Semi-Joins (aplica en BDD en donde el recurso más costoso es el uso de la Red)
 - Re-ordena operaciones para reducir cardinalidades de resultados parciales.
- *Optimizador basado en reducción de tiempos*
 - Estático
 - La consulta es optimizada antes de su ejecución
 - Dificultad para estimar el tamaño de resultados intermedios.
 - Requiere el uso de estadísticas adecuadas para generar estimaciones correctas.
 - Dinámico
 - La optimización se realiza al ejecutar la consulta.
 - Ofrece resultados intermedios exactos.
 - No requiere el uso de estadísticas, aunque se suele emplearlas para determinar las operaciones que se ejecutarán al principio y después se emplean los resultados obtenidos para decidir los siguientes pasos.
 - Híbrido
 - Inicia con optimización estática.
 - Si el resultado obtenido en una primera operación sobre – pasa un umbral establecido, la consulta es optimizada de forma dinámica.
 - *Optimizadores que explotan fragmentos replicados.*
 - El uso de replicación permite además de aumentar el grado de confiabilidad, incrementar el desempeño de lecturas (al seleccionar el sitio más cercano).
 - Los optimizadores pueden hacer uso de datos replicados para mejorar los costos y tiempos de procesamiento.

3.3. ETAPAS DENTRO DEL PROCESAMIENTO DE CONSULTAS.

Existen variantes, pero en general el siguiente esquema muestra las etapas del procesamiento de una consulta distribuida en donde la estrategia es construida desde un solo sitio y no se hace uso de datos replicados.



- Las 3 primeras etapas se ejecutan en el sitio que genera la estrategia de ejecución.
- Las últimas 2 etapas se ejecutan en cada uno de los nodos de donde se obtendrán los datos.

En el tema 4 se revisarán las primeras 3 etapas y en el capítulo 5 las últimas 2.