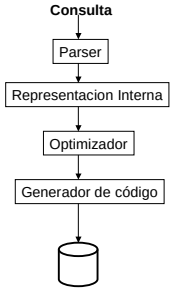
Procesamiento de Consultas

Procesamiento de Consultas

- **Heterogeneidad de los datos**:
  Desafío de desarrollar lenguajes que puedan ser usados para formular consultas que envuelvan modalidades múltiples de datos
  (ej. registros,texto, imágenes,video, sonido).

Tema:
Procesamiento de consultas sobre registros.

Procesamiento de Consultas en BD Centralizadas

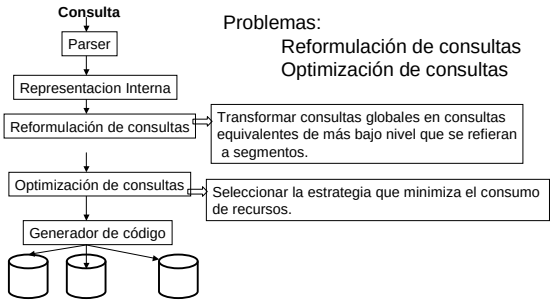


Query trees
Algebra relacional

Reglas Heuristicas
Modelo de Costo

(Costo de:
acceso a los datos,
acceso al almacenam secundario,
realizar computos.)

Procesamiento de Consultas en BD Distribuidas



Problemas:
Reformulación de consultas
Optimización de consultas

Transformar consultas globales en consultas equivalentes de más bajo nivel que se refieran a segmentos.

Seleccionar la estrategia que minimiza el consumo de recursos.

Procesamiento de Consultas en BD Distribuídas

- Datos compartidos a través de una red de nodos donde cada nodo es una BD Homogénea.
  - Localización de los datos para formular reglas heuristicas.
  - Descomposición
    - Consultas en paralelo en cada nodo
  - Costo de transferencia de los datos sobre la red.
    - Reducir la cantidad de datos a transferir (ej. Semi-join)

Procesamiento de Consultas en BD Heterogéneas

Diferencias con BDDistribuidas:

- **Diferencias de capacidades en cada fuente**
  - Las fuentes pueden ser sistemas legados con interfaces a los datos muy limitadas.
  - Aún si todos los datos están almacenados en DBMS tradicionales estos pueden proveer acceso limitado por seguridad o performance.
  - Pueden tener procesamientos de consultas adicionales

Procesamiento de Consultas en BD Heterogéneas (II)

Diferencias con BDDistribuidas:

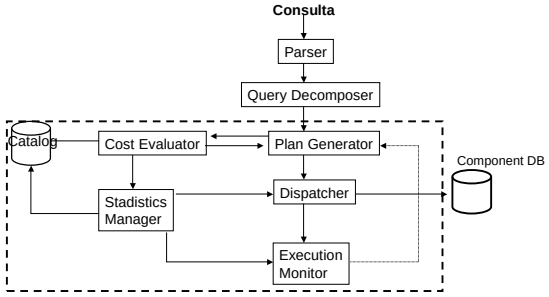
- **Información sobre las fuentes y la red no disponible**
  - Costo local de las consultas es desconocido
  - Difícil de estimar las estadísticas sobre los datos
  - Costos de transferencias impredecibles

Procesamiento de Consultas en BD Heterogéneas

Problema de la Autonomía:

- Control completo sobre los datos locales
- Sitios libres de unirse o no al sistema (autonom. de comunicación)
- Optimizador de consultas locales (autonom. de diseño)
  (Hace que la estadistica global de costo quede desactualizada)
- Cooperan a través de la interface, no hay oportunidad de cooperación a bajo nivel (no semi-join)

Procesamiento de Consultas en BD Heterogéneas



Reformulación de Consultas (Query Decomposer)

Problema:

Reformular la consulta global como consultas en las fuentes de información.

Dados:

- Una consulta Q en términos del esquema mediado (virtual approach)
- Descripciones de las fuentes

Encontrar:

Una consulta Q' que usa solo la información en las fuentes tal que:
Q' provee todas las posibles respuestas a Q usando las fuentes.

Descripción de las fuentes

Distintos encares de especificación:

- Global As View (GAV):
  El esquema mediado es definido como vistas sobre los esquemas fuentes.

  Por cada relación R en el esquema mediado, escribimos una consulta sobre las relaciones fuentes especificando como obtenemos las tuplas de R desde las fuentes.

Proyectos:

TSIMMIS (Stanford),
HERMES (U. Maryland),
DISCO (INRIA)

Descripción de las fuentes (II)

- Local As View (LAV):
  Los esquemas fuentes son definidos como vistas sobre el esquema mediado.

  Por cada fuente de información S escribimos las relaciones en el esquema mediado que describen las tuplas encontradas en S.

Proyectos:

Information Mainfold (AT & T)
Occam (U of Washington)
Info Master (Stanford)

Comparación GAV vs. LAV

- Global As View (GAV):

  Reformulación de consultas es muy simple

- Local As View (LAV):

  Es simple agregar o remover fuentes porque la descripción de las fuentes no necesitan tener en cuenta las posibles iteracciones con otras fuentes.

Reformulación de Consultas

Realizar reformulación de consultas usando vistas:

- Dado un conjunto de definición de vistas V1, V2, ...Vn y una consulta Q, encontrar una consulta Q' que:
  - use solo las vistas V1, V2 ...Vn
  - que esté contenida en Q y soporte tantas respuestas de Q como son posibles desde las vistas.

Algoritmos de reformulación de consultas

- Su complejidad depende de:
  - Del lenguaje usado en la consulta y en las definiciones de vistas
  - Si se hace reformulación *equivalente* de consultas o reformulación *conteniendo* la consulta.
- Técnicas:
  - Aplicar un conjunto de reglas de re-escritura o reformulación hasta que las vistas estén explícitamente en la consulta.

  [Levy, Rajaraman & Ullman 1996]

Optimización de consultas

Las propiedades del álgebra relacional nos facilitan asegurar la correctitud de la reformulación de la consulta.

Sin embargo, producir una ejecución eficiente es más elaborado
⇒ El problema de optimización de consultas:

Transformar una consulta declarativa en un programa imperativo equivalente de mínimo costo.

El programa imperativo es un Plan de Ejecución de la Consulta (QEP):
árbol de operadores en álgebra que selecciona la estrategia que minimiza el consumo de recursos.

Objetivos de la optimización de consultas

- Minimizar costos
- En un DDBMS el costo total incluye:
  CPU + I/O + comunicación

Se reduce a un enfoque donde:
1 Minimizar costos de comunicación
2 Reducir al problema de bases de datos centralizadas:
minimizar localmente CPU + I/O

Usar reglas de transformaación basadas en propiedades del álgebra relacional:
Simplificar predicados, detectar subexpresiones comunes, aplicar selecciones y proyecciones tan pronto como sea posible.

Optimización de Consultas en BD Heterogéneas

Diferencias con BDDistribuidas:

- **Información sobre las fuentes y la red no disponible**
  - Costo local de las consultas es desconocido
  - Difícil de estimar las estadísticas sobre los datos
  - Costos de transferencias impredecibles

Soluciones propuestas para optimización

"Query Sampling"

Clasifica las consultas en la bd local en *clases.*
[Zhu and Larson, 1994]

Criterios de clasificación según caracteristicas :
de la sintaxis de la consulta, de las tablas (cardinalidad, índices),
del soporte de métodos de acceso de la BD local

Se escriben consultas de muestra para cada clase de consultas

Estas consultas son realizadas en la BD local y su costo observado y almacenado

Query Sampling

PARA ESTIMAR EL COSTO DE UNA CONSULTA:

1) identificar la clase de consulta a la cual pertence

2) La fórmula de costo observada para esa clase es usada para estimar el costo de la consulta.

PROBLEMAS:

Las muestras de los costos para cada clase se tienen que realizar a intervalos no muy grandes pues sino pueden dejar de ser relevantes debido a que las bd locales estan libres de modificar cualquier caracteristica de las consultas

Pero si es realizado muy a menudo la performance del MDBMS puede estar afectado a causa de este costo extra.

Query Probing and Piggyback

Variante del Query Sampling:
Usa algunas consultas especiales (Probing queries)

De estas consultas obtiene:
catálogo global
información estadística (ej, cardinalidad de las tablas) e
información sobre el esquema (ej, índices disponibles).

Esta información es usada para estimar el costo de la consulta.

Mismos problemas que en Query Sampling

El modelo de costo de Regresión y Calibración

"Calibration and Regression Cost Model" [Gardarin *et. al* , 1996]

Query Sampling: Un costo por clase de consulta.
Calibration and Regression Cost Model: coeficientes para c/u de las variables que contribuyen en la formula de costo usando una BD calibrada.

- Variables que contribuyen:
  Variables relevantes al costo por metodo de Regresión

Comienza agregando de a una varibale por vez para hallar las variables relevantes al costo para cada clase de consulta y luego por regresión va eliminando de a una. Combinando estos dos metodos la mayor posibilidad cuando el algoritmo termina es que obtiene las variables mas relevantes.

- BD Calibrada: consiste de relaciones de varios tamaños con datos producidos deterministicamente tal que cada columna de las tablas tiene una distribucion de datos especifica.

Proactive Approach

La idea en este enfoque es la de sistematicamente optimizar el QEP cada vez que hay nueva información.

No genera un QEP en tiempo de ejecucion.
Por el contrario: determina el próximo paso en la secuencia de ejecución solo despues que el paso previo fue completado.

El optimizador de consultas monitorea la ejecución de las subconsultas para determinar:

- tiempo de respuesta de la subconsulta
- tamaño de los resultados intermedios

Problemas del Proactive Approach

- Ventajas:
  Provee al optimizador de consultas con datos más confiables en términos de los costos de ejecución.
- Desventajas:
  Costo extra demasiado grande para la generación dinámica de los QEP
- Versión refinada:
  Generar el QEP en tiempo de compilación y modificarlo solo si los costos de las subconsultas son mucho mas altos que los esperados.

El enfoque STAR

"Strategy Alternative Rules" (STAR) [Hass *et al* 1997]

Consiste de un conjunto de reglas.
Usando estas reglas se construyen:
- todos los *posibles* QEP para las BD locales,
- sin importar el costo

De todas las posibles el optimizador selecciona la de menor costo ⇒ Produce recarga en el trabajo de encontrar la mas barata

Conclusiones

- Necesidad de mejores QEP
- Necesidad de mejores modelos de costos
- Identificaar clases interesantes de consultas y contextos de aplicación
- Testear los prototipos en aplicaciones reales
- Benchmarks

Bibliografía

- **ZhuQ. And Larson P 1994**

A query sampling method for estimating local cost parameters in a multidatabase system.
In Proceedings of the 10 th IEEE International Conf. On Data Engineering, pages 144-153, Houston, Texas.

- **Gardarin G., Sha F. and Tang Z. 1996**

Calibrating the query optimizer cost model of IRO-DB, an object-oriented federated database system.
In Proceedings of 22nd International VLDB Conference, Bombay.

- **Haas L., Kossmann D., Wimmers E. and Yang J. 1997**

Optimizing queries across fdiverse data sources.
In proceedings of the 23th International VLDB Conference, Athens.