Capítulo 4

Processamento de Consultas

Tem como função:

- Transformar uma query de alto nível (cálculo relacional) numa query equivalente de baixo nível (álgebra relacional)
- Escolher uma estratégia de processamento de queries com menor custo de recursos computacionais
- Buscar entre várias transformações equivalentes a mais correta

No caso da BD distribuída, a consulta deve ser estendida com operações de comunicação e otimização.
Existem 3 passos para atingir:

- Analisar e Traduzir: Verificar a sintaxe e as relações
  Traduzir (decompor) numa expressão de álgebra relacional equivalente
- Otimização: Plano de avaliação ideal (localização dos dados)
- Execução: Execução do plano e avaliação

O sucesso de SGBDR (Sistemas de Gestão de Base de Dados Relacionais) é devido a:

- Linguagem de queries declarativas e fácil
- Tecnologias de processamento de queries avançada

A transformação deve atingir:

- Correção: Fácil de alcançar através do mapeamento bem definido do cálculo relacional para álgebra relacional
- Eficiência: É difícil de alcançar pois é difícil selecionar a estratégia de execução que minimiza o consumo de recursos

Processos de Consultas Distribuídas

- **Objetivos:**
  - Dar ao utilizador a impressão de que a query é realizada numa única BD
  - Transformar uma query de alto nível definida para um BD distribuída (que parece ser uma única BD) numa estratégia de execução eficiente (expressa em linguagem de baixo nível) a ser executada em BD locais

- **Otimização de Consultas:**
  - Para uma mesma consulta de alto nível podem existir muitas estratégias de execução diferentes - aquela que minimiza o consumo de recursos deve ser a escolhida

- **Medidas de Consumo:**
  - Minimizar o custo de: I/O + CPU + custo de comunicação
    - redes de comunicação mais rápidas (LAN e WAN)

- **Operações da álgebra relacional:**
  - A álgebra relacional é a base para expressar o processamento de uma consulta
    - A complexidade dos operadores da álgebra relacional afetam diretamente o tempo de execução de uma query e ditam alguns princípios úteis ao processamento
    - A complexidade é definida pela cardinalidade
    - Os operadores mais seletivos devem ser executados primeiro (reduzem a cardinalidade)
    - Operadores devem ser ordenados pela complexidade de forma crescente (produto cartesiano deve ficar para último

Problemas no processo de Otimização das Queries

Problemas a considerar na otimização das queries:

- Técnica: procura exaustiva, heurística, híbrida
- Tempo: quando a query é otimizada (estática, dinâmica)
- Estatísticas: número de tuplos, o seu tamanho e o número de colunas
- Decisão dos sites: centralizado, distribuído, misturado (distribuição de como os elementos estão dispostos)
- Topologia da Rede: LAN ou WAN
- Fragmentos replicados
- Uso de *semijoins*: reduz o tamanho de *joins* e o custo de comunicação

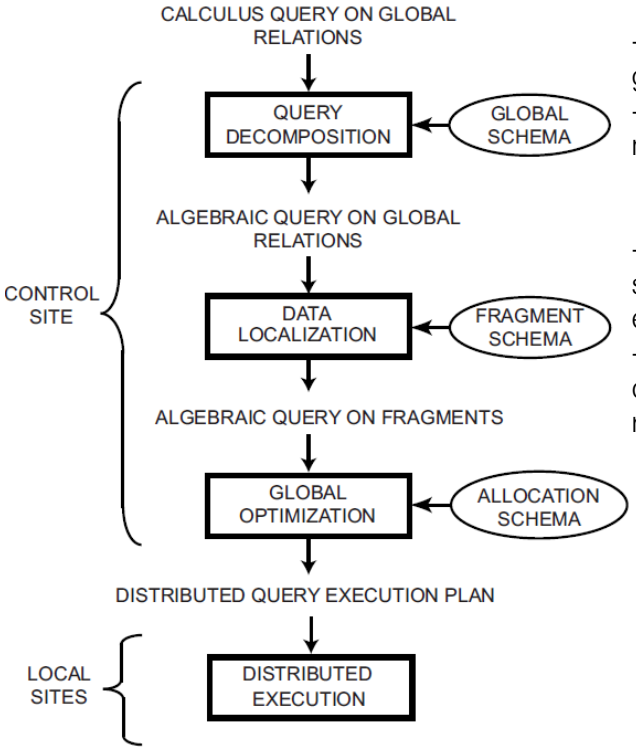
Técnica	- Procura exaustiva - elevado custo para manter mas ótimo - Heurística - baixo custo mas não ótimo - Híbrida - é construída uma query estática e a adaptação é abaliado em tempo de execução

Tempo	- Estático
	  - A otimização da query ocorre em tempo de compilação
	  - O custo pode ser amortizado por conta das várias execuções da query
	  - É apropriado para usar com o método de pesquisa exaustiva
	- Dinâmico
	  - A otimização da query ocorre em tempo de execução
	  - É possível a qualquer ponto da execução escolher a melhor estratégia de execução considerando o conhecimento preciso dos resultados
	  - Desvantagem: A otimização fica cara
	  - É apropriado para usar com o método de pesquisa híbrido

Decisão dos Sites	- Centralizado
	  - Um único site toma a decisão
	  - Requer o conhecimento de toda a BD distribuída
	- Distribuído
	  - Vários sites tomam a decisão
	  - Requer apenas informações locais
	- Híbrida
	  - Um único site toma a decisão mais importante e os restantes tomam decisões locais

Estatística	- As estatísticas para a otimização de consultas são definidas com base nos fragmentos, incluindo a cardinalidade e tamanho dos fragmentos, bem como o tamanho e o número de valores distintos de cada atributo - Atualizações periódicas das estatísticas levam a uma “re-otimização” no caso da otimização estática
Fragmentos Replicados	Para fins de fiabilidade é útil ter fragmentos replicados - Alguns algoritmos de otimização exploram a existência de fragmentos replicados em tempo de execução visando diminuir o custo de comunicação - O algoritmo torna-se mais complexo porque aumenta o número de possibilidades de estratégia em execução
Uso de *semijoins*	Quando o principal componente de custo é a comunicação, um *semijoin* é útil para melhorar o processamento, pois reduz o tamanho dos dados a serem trocados pela rede. Porém, pode resultar num aumento de troca de mensagens e do tempo de processamento local

Camadas de Processamento de Consulta



- A entrada é uma consulta definida sobre relações globais (a distribuição fica transparente).
- Os 3 primeiros níveis mapeiam a query de entrada num plano de execução de query distribuída.
    - A decomposição da consulta e a localização dos dados correspondem à reescrita da query.
- As 3 primeiras atividades são executadas por um site de controlo central e usam informações sobre o esquema armazenado no diretório global.
- O 4º nível é responsável pela execução da query distribuída (o plano é executado e a resposta é retornada ao utilizador)
    - isto é feito pelos sites locais e pelo site de controlo

Decomposição de consultas

- Decomposição da query de cálculo algébrico numa consulta algébrica sobre relações globais
- As técnicas utilizadas por essa camada são as de um SGBD centralizado

Nomalização	A query de cálculo é reescrita de forma normalizada
Análise	A query normalizada é analisada semanticamente, de forma a que as queries incorretas sejam rejeitadas
Simplificação	As queries corretas são simplificadas, eliminando predicados redundantes
Restruturação	A query de cálculo é reestruturada para uma query algébrica

Localização dos Dados

- Localiza os dados e determina os fragmentos que estão envolvidos na query e transforma a query distribuída numa query sobre fragmentos

Entrada	Query algébrica sobre relações distribuídas
Objetivo	Localizar dados com o uso de informações de distribuição de dados
Resultado	Determina que fragmentos estão envolvido na query e transforma a query distribuída numa query sobre fragmentos

Otimização de Queries Globais

- Encontrar a melhor estratégia de execução possível. A estratégia consiste em ordenar as operações incluindo as de comunicação para minimizar o custo computacional

Entrada	Query algébrica sobre os fragmentos
Objetivo	- Encontrar a melhor estratégia de execução possível - A estratégia de execução pode ser descrita como operações de álgebra relacional e primitivas de comunicação para transferência de dados entre sites - A estratégia consiste em ordenar as operações incluindo as de comunicação, para minimizar o custo computacional
Resultado	Plano de execução de consulta distribuída

Execução Distribuída

- Otimização da query usando um repositório local, usando algoritmos de sistemas centralizados

Entrada	Operações de álgebra relacional
Objetivo	Otimização da query usando um repositório local
Resultado	A otimização local usa algoritmos de sistemas centralizados