

Capítulo 4

Processamento de Consultas

Visão Geral do Processamento de Consultas

O grande sucesso dos SGBDs tem origem, principalmente, na facilidade que eles oferecem para organizar e acessar os dados de maneira transparente, ou seja, sem preocupações com a localização física dos dados.

Um componente fundamental dos SGBDs é o processador de consultas!

Tem como objetivo:

- Transformar uma consulta de alto nível (cálculo relacional - declarativa) numa consulta equivalente de baixo nível (álgebra relacional - procedural)

Análise e Tradução	Otimização	Execução
<ul style="list-style-type: none">- Verificar a sintaxe e verificar as relações- Tradução (decomposição) numa expressão de álgebra relacional equivalente	<ul style="list-style-type: none">- Plano de avaliação ideal (local de dados)	<ul style="list-style-type: none">- Execução da avaliação
<ul style="list-style-type: none">- Escolher a estratégia de processamento de consulta com o menor custo de recursos computacionais- Procurar entre muitas transformações equivalentes e corretas		
<ul style="list-style-type: none">- No caso de Base de Dados Distribuídas, a consulta deve ser estendida como operações de comunicação e otimização		

Sucesso e Eficiência:

- O sucesso do Sistema de Gestão de BD Relacionais (SGBDR) é devido:
 - Uma linguagem de fácil consulta declarativa
 - Consulta avançada tecnologias de processamento
- Transformação:
 - Correção: fácil de alcançar
 - Eficiência:
 - É difícil de alcançar
 - É um dos objetivos mais importantes

Distribuição:

- O processamento de consultas distribuídas é mais difícil:
 - Fragmentação e replicação
 - Comunicação: custos e distribuição
 - Execução paralela

EXEMPLO

Empregados - EMP (ENO, ENAME, TITLE)

Empregados e Projetos - ASG (ENO, PNO, RESP, DUR)

Encontre os nomes dos empregados que gerem um projeto

```
SELECT ENAME
  FROM EMP, ASG
 WHERE EMP.ENO = ASG.ENO
   AND RESP = 'Manager'
```

Transformação das consultas equivalentes em álgebra relacional

$$\Pi_{ENAME}(\sigma_{RESP='Manager'} \wedge EMP.ENO = ASG.ENO (EMP \times ASG))$$

and

$$\Pi_{ENAME}(EMP \bowtie_{ENO} (\sigma_{RESP='Manager'} (ASG))) \leftarrow \text{evita o produto cartesiano extenso}$$

EXEMPLO de Base Dados Distribuídas

Seja a consulta: “Liste os nomes dos empregados que são gerentes”

$$\Pi_{ENAME}(EMP \bowtie_{ENO} (\sigma_{RESP='Manager'} (ASG)))$$

A informação está em Fragmentação Vertical.

As relações EMP e ASG são fragmentadas como se segue:

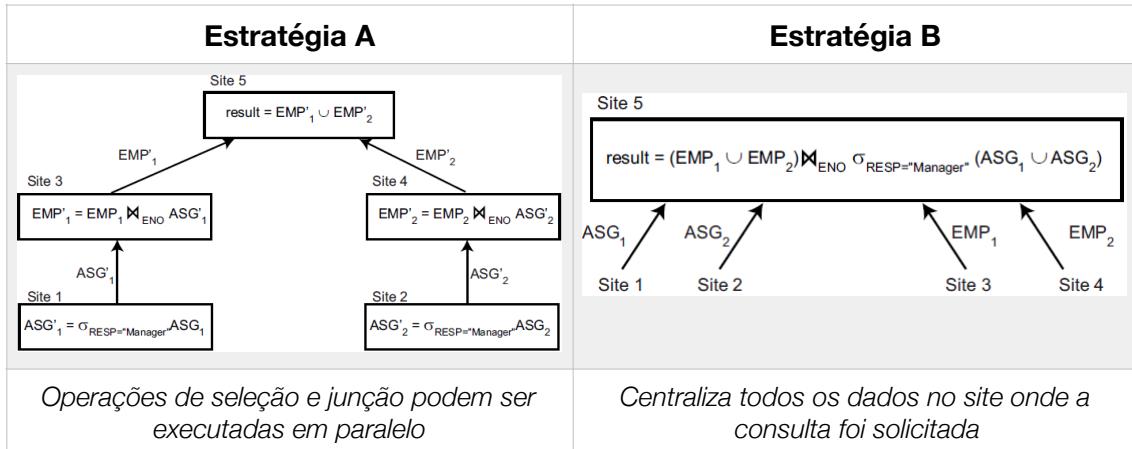
$$EMP_1 = \sigma_{ENO \leq E_3} (EMP) \leftarrow \text{Armazenado no site 1}$$

$$EMP_2 = \sigma_{ENO > E_3} (EMP) \leftarrow \text{Armazenado no site 2}$$

$$ASG_1 = \sigma_{ENO \leq E_3} (ASG) \leftarrow \text{Armazenado no site 3}$$

$$ASG_2 = \sigma_{ENO > E_3} (ASG) \leftarrow \text{Armazenado no site 4}$$

E o resultado da consulta é esperado no site 5!



Custo

Acesso ao tuplo: 1 unidade

Transferência do tuplo: 10 unidades

Relação EMP contém 400 tuplos

Relação ASG contém 1000 tuplos

Os dados são uniformemente distribuídos em cada site

Existem 20 empregados que são gerentes na relação ASG

1. Produce ASG' by selecting ASG requires $(10 + 10) * tupacc$	=	20
2. Transfer ASG' to the sites of EMP requires $(10 + 10) * tuptrans$	=	200
3. Produce EMP' by joining ASG' and EMP requires $(10 + 10) * tupacc * 2$	=	40
4. Transfer EMP' to result site requires $(10 + 10) * tuptrans$	=	200
The total cost is		<u>460</u>

Custo total da estratégia A

Custo total da estratégia B

1. Transfer EMP to site 5 requires $400 * tuptrans$	=	4,000
2. Transfer ASG to site 5 requires $1000 * tuptrans$	=	10,000
3. Produce ASG' by selecting ASG requires $1000 * tupacc$	=	1,000
4. Join EMP and ASG' requires $400 * 20 * tupacc$	=	8,000
The total cost is		<u>23,000</u>

Objetivos do processamento de consultas distribuídas

- Dar ao utilizador a impressão de que a consulta é realizada numa única base de dados
- Transformar uma consulta em alto nível definida para uma BD Distribuído (que parece ser um único BD) numa estratégia de execução eficiente (expressa numa linguagem de mais baixo nível) a ser executada nas base de dados locais
- Otimização das consultas

Para uma mesma consulta de alto nível podem existir muitas estratégias de execução diferentes – aquela que minimiza o consumo de recursos deve ser escolhida
- Medidas de consumo

Custo total (CPU, I/O e comunicação): soma de todos os tempos que incidem no processamento das operações de consulta em diversos sites e na comunicação entre eles.

Tempo de resposta da consulta: tempo decorrido para execução da consulta

 - As operações podem ser executadas em paralelo em diversos sites: tempo de resposta de uma consulta pode ser menor que o custo total
- Objetivo da otimização de consultas distribuídas reduz-se ao problema de minimizar os custos de comunicação de forma geral

Otimização local pode ser feita de forma independente

 - Atualmente buscamos uma combinação ponderada dos 3 tipos de custo (CPU, I/O e comunicação)

Redes de comunicação estão cada vez mais rápidas

Complexidade de operações da álgebra relacional

Álgebra relacional é a base para expressar o processamento duma consulta

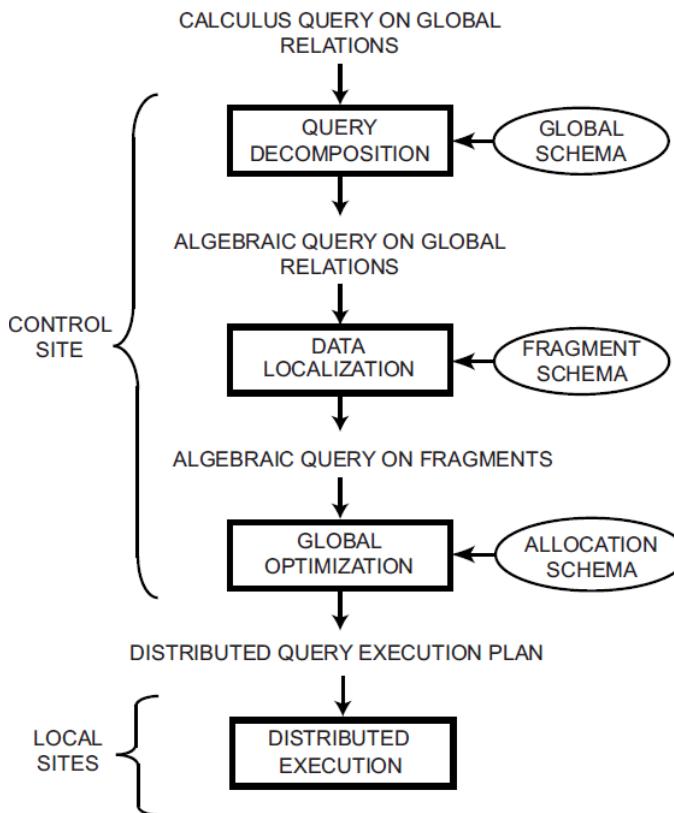
- A complexidade dos operadores da álgebra relacional afetam e ditam alguns princípios úteis ao processamento
 - A complexidade é definida em termos da cardinalidade das relações
 - Os operadores mais seletivos (reduzem a cardinalidade) devem ser executados primeiro
 - Operadores devem ser ordenados pela complexidade de forma crescente (o produto cartesiano deve ficar para o final)

Caracterização de processadores de consulta

- Características que podem ser usadas para comparar os processadores de consulta
 - Aplicam-se a BD centralizadas e BD distribuída
 - Linguagens
 - Tipos de Otimização
 - Momento da Otimização
 - Estatísticas
 - Aplicam-se apenas a BD distribuídas
 - Sites de decisão
 - Exploração da Topologia da Rede
 - Exploração de fragmentos replicados
 - Uso de *semijoins*

Linguagens	Tipos de otimização
<ul style="list-style-type: none"> • SGBDs relacionais oferecem muitas possibilidades de otimização <ul style="list-style-type: none"> - Pode ser baseada no cálculo relacional ou na álgebra relacional - Exige uma fase adicional para decompor uma consulta expressa em cálculo relacional para álgebra relacional • SGBDs de objetos utilizam o cálculo de objetos sendo uma extensão do cálculo relacional <ul style="list-style-type: none"> - Decomposição em álgebra de objetos também é necessária • Linguagens para BD Distribuídas <ul style="list-style-type: none"> - Algumas forma interna da álgebra relacional ampliada com primitivas de comunicação 	<ul style="list-style-type: none"> • Buscar em todo o espaço de solução (calculando o custo de execução de cada estratégia) para selecionar aquela com custo mínimo • Pode incorrer num custo de processamento significativo • Outras soluções podem ser usadas a fim de encontrar uma boa solução sem ser necessariamente a melhor solução <ul style="list-style-type: none"> - Utilizar heurísticas para diminuir o espaço de possibilidades
Momento da Otimização	
<ul style="list-style-type: none"> • Estática (antes da execução da consulta) <ul style="list-style-type: none"> - Ocorre em tempo de compilação - O custo elevado pode ser amortizado por conta das várias execuções da consulta - Usa apenas estatísticas da base de dados - Não é possível considerar informações sobre as relações intermediárias - estas informações não são disponíveis! • Dinâmica <ul style="list-style-type: none"> - Ocorre em tempo de execução - Podendo em qualquer ponto da execução escolher a melhor estratégia de execução considerando o conhecimento preciso dos resultados - nem sempre as estatísticas são precisas! - Pode fazer uso de informações sobre as relações intermédias - <u>Desvantagem:</u> A otimização é uma tarefa cara e deverá ser realizada cada vez que a consulta for executada 	
Estatísticas	Sites de Decisão
<ul style="list-style-type: none"> • Otimização estática: o tamanho das relações intermediárias é estimado com base nas informações estatísticas • Otimização dinâmica: faz uso de estatísticas para determinar os operadores a serem executados primeiro • As estatísticas para a otimização de consultas são definidas com base nos fragmentos, incluindo cardinalidade e tamanho dos fragmentos, bem como o tamanho e o número de valores distintos de cada atributo • Atualização periódica das estatísticas leva a uma "re-otimização" no caso da otimização estática 	<ul style="list-style-type: none"> • A decisão da estratégia a ser aplicada pode ser tomada por um ou vários sites • Abordagem de decisão centralizada (tipicamente usada): <ul style="list-style-type: none"> - Mais simples, um único site toma a decisão - Requer o conhecimento de toda a base de dados distribuída • Abordagem distribuída: <ul style="list-style-type: none"> - Vários sites tomam a decisão - Requer apenas informação locais • Abordagens híbridas: <ul style="list-style-type: none"> - Um único site toma a decisão mais importante - O restante toma decisões locais
Exploração da topologia de rede	Exploração de fragmentos replicados
<ul style="list-style-type: none"> • Em redes remotas, a função de custo pode estar restrita a comunicações de dados, que é o fator dominante • A otimização de consultas pode ser dividida em dois problemas: <ul style="list-style-type: none"> - Execução global, baseada na comunicação entre sites - Execução local, baseado em algoritmo de consulta centralizado • Em redes locais, os custos de comunicação são mais reduzidos <ul style="list-style-type: none"> - Nesse caso, o processador de consultas distribuídas pode aumentar o paralelismo na execução das consultas 	<ul style="list-style-type: none"> • Consultas distribuídas definidas sobre relações globais são mapeadas em consultas sobre os fragmentos físicos <ul style="list-style-type: none"> - Localização: É o processo de localizar os dados (fragmentos) envolvidos numa consulta • Para fins de confiabilidade é útil ter fragmentos replicados <ul style="list-style-type: none"> - 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 estratégias de execução
Sincronização da Otimização	Uso de semi-join
<ul style="list-style-type: none"> • Híbrida <ul style="list-style-type: none"> - Procura oferecer as vantagens da estática - Utiliza a abordagem dinâmica quando deteta em tempo de execução uma diferença grande entre o custo previsto (baseado nas estatísticas) e as relações intermediárias reais 	<ul style="list-style-type: none"> • Quando o principal componente de custo é a comunicação, um <i>semi-join</i> é útil para melhorar o processamento, pois reduz o tamanho dos dados a serem trocados na rede • Porém, pode resultar no aumento de troca de mensagens e do tempo de processamento local ! <ul style="list-style-type: none"> Um <i>semi-join</i> da relação R definida sobre o conjunto de atributos A pela relação S definida sobre o conjunto de atributos B é o subconjunto de tuplos de R que participam da junção de R com S

Camadas do processamento de consultas



- A entrada é uma consulta definida sobre relações globais (a distribuição fica transparente)
- Os três primeiros níveis mapeiam a consulta de entrada num plano de execução de consulta distribuída
 - A decomposição da consulta e a localização dos dados correspondem a reescrita da consulta
- As três primeiras atividades são executadas por um site de controlo central e usam informações sobre o esquema armazenadas no diretório global
- O quarto nível é responsável pela execução da consulta 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 consulta de cálculo algébrico numa consulta algébrica sobre relações globais
- As técnicas utilizadas por essa camada são as de um SGBD centralizado
- 4 etapas sucessivas:
 - A consulta de cálculo é reescrita de forma normalizada adequada a manipulação subsequente
 - A consulta normalizada é analisada semanticamente, de forma que as consultas incorretas sejam detetadas e rejeitadas o quanto antes
 - As consultas corretas são simplificadas, eliminando predicados redundantes
 - A consulta em cálculo é reestruturada para uma consulta algébrica

Localização de dados

- Entrada: Consulta algébrica sobre relações distribuídas
- Localizar dados com o uso de informações de distribuição de dados
- Determinar que fragmentos estão envolvidos na consulta e transformar a consulta distribuída numa consulta sobre fragmentos

Otimização de consultas globais

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

Otimização de consultas locais

- Entrada: Operações de álgebra relacional da estratégia definida
- Executado por todos os sites que têm fragmentos envolvidos na consulta
- Cada subconsulta realizada num site, chamada consulta local, é otimizada com uso do esquema local
- A otimização local emprega os algoritmos de sistemas centralizados

Execução da consulta

- Entrada: Consultas algébricas em fragmentos
- Otimização da consulta usando o repositório local
- Otimização local usa algoritmos de sistemas centralizados

Conclusão

- Existem várias formas de executar uma consulta!
- O processador de consultas é o responsável por encontrar uma forma eficiente de execução de uma consulta
- O processador de consultas utiliza dados do sistema para estimar o custo de cada plano de execução
- É muito importante avaliar e otimizar as consultas, pois o desempenho do sistema pode ser drasticamente afetado dependendo do plano de execução escolhido