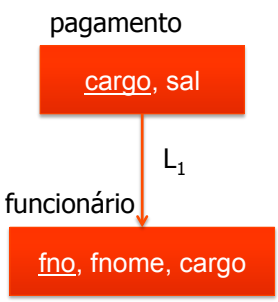
Projeto de Bancos de Dados Distribuídos (aula 02)

IF694 – BD Distribuídos e Móveis
Bernadette Farias Lóscio
bfl@cin.ufpe.br

Centro de Informática UFPE — Universidade Federal de Pernambuco — CIn.ufpe.br

Fragmentação horizontal derivada

- **Uma fragmentação horizontal derivada é definida sobre uma relação membro de um vínculo de acordo com uma operação de seleção especificada sobre sua proprietária**
- **Exemplo:**



Exemplo:

Proprietária(L_1) = pagamento
Membro(L_1) = funcionário

Na fragmentação horizontal derivada, um fragmento de funcionário seria definido de acordo com uma seleção em pagamento

Fragmentação horizontal derivada

- **Em uma fragmentação horizontal derivada, os fragmentos são definidos considerando operações de semijunção**
  - Deseja-se fazer o particionamento de uma relação membro de acordo com uma relação proprietária, porém o resultado da fragmentação dever ser definido somente sobre os atributos da relação membro
  - Um semijunção da relação ***R*** definida sobre o conjunto de atributos ***A*** pela relação ***S*** definida sobre o conjunto de atributos ***B*** é o subconjunto das tuplas de ***R*** que participam da junção de ***R*** com ***S***

Fragmentação horizontal derivada

- **Dada uma ligação *L* onde *proprietária*(*L*)=*S* e *membro*(*L*)=*R*, Os fragmentos horizontais derivados de R são definidos como:**
  - R_i = *R semijoin* S_i, 1 ≤ *i* ≤*w*, onde *w* é o número de fragmentos definidos sobre R e
  - *Si* = σ_{Fi} (*S*), onde F_i é a fórmula segundo a qual o fragmento horizontal primário S_i é definido

Fragmentação horizontal derivada - exemplo

- **Dada a ligação L_1 onde:**
  - Proprietária(L_1) = Pagamento
  - Membro(L_1) = Funcionário
- **A relação funcionário pode ser particionada da seguinte forma:**
  - Funcionário1 = Funcionário *semijoin* Pagamento1
  - Funcionário2 = Funcionário *semijoin* Pagamento2
  - onde:
  - Pagamento 1 = $\sigma_{salário \leq 30000}$ (*Pagamento*)
  - Pagamento 2 = $\sigma_{salário > 30000}$ (*Pagamento*)

Fragmentação horizontal

- **É importante notar que é possível ter mais de uma fragmentação candidata para uma relação**
- **A escolha final do esquema de fragmentação pode ser um problema de decisão solucionado durante a alocação**

Fragmentação Vertical

- **Processo que divide uma relação em colunas, agrupando em cada fragmento parte dos atributos existentes na relação original**
- **Replicação da chave da tabela original nos fragmentos gerados**
  - Fato que permite a reconstrução da relação global

Objetivo

- **Particionar uma relação em um conjunto de relações menores, para que muitos dos aplicativos do usuário possam atuar apenas sobre um fragmento**
- **Caso uma aplicação precise de dados presentes em mais de um fragmento: fragmentação não benéfica (necessidade de junções, operação muito custosa)**

Complexidade

- **Processo inerentemente mais complexo que a fragmentação horizontal;**
  - Fragmentação horizontal.: n predicados – 2^n predicados minterm possíveis, tirando ainda os que não fazem sentido;

Complexidade

- **Fragmentação vertical: A quantidade de fragmentos possíveis é dada por B(m), onde m são os atributos não chave e B é o número de Bell (número de partições de um conj. de n elementos)**
  - Ex.:
  - B(4) = 15
  - B(10) ≈ 115.000
  - B(15) ≈ 109
  - B(30) ≈ 1023
- **Esses valores indicam que não vale a pena tentar obter soluções ótimas, sendo necessário recorrer à abordagem heurística**

Abordagens Heurísticas

- **Há dois tipos de abordagens heurísticas pra fragmentação vertical:**
  - Agrupamento: Começa atribuindo cada atributo a um fragmento e vai juntando (Join) até achar os que satisfazem
  - Divisão*: Começa com uma relação e define como particionar de acordo com o comportamento de acesso aos atributos
- ***se ajusta de forma mais natural dentro da metodologia de projeto top-down**

Fragmentação vertical

- **Como a fragmentação vertical insere os atributos em um único fragmento que, em geral, são acessados em conjunto, surge a necessidade de algumas medidas que definam com maior precisão a noção de conjunto**

Processo de Fragmentação Vertical

- **Afinidade de atributos: indica a proximidade dos relacionamentos entre atributos**
- **Como obter esse valor a partir dos dados do BD?**
- **Pra isso, deve-se saber o que vem a ser "valor de uso do atributo":**
  - Seja Q = {q1,q2, ... , qm} consultas do usuário na relação R (A1,A2, ... , An)
  - Para cada consulta qi e cada atributo Aj, associamos um valor de uso do atributo, denotado por uso(qi,Aj) e definido como se segue:
    - uso(qi,Aj) = 1, se o atributo Aj é referido pela consulta qi
    - uso(qi,Aj) = 0, caso contrário

Processo de Fragmentação Vertical

- **Exemplificando:**

q_1: encontrar orçamento de um projeto pelo ID
"SELECT orcamen FROM proj WHERE pno = valor"

q_2: encontrar nomes e orçamentos dos projetos
"SELECT nome, orcamen FROM proj"

q_3: encontrar nomes dos projetos em uma dada cidade
"SELECT nome FROM proj WHERE loc = valor"

q_4: encontrar orçamentos dos projetos em uma dada cidade
"SELECT orcamen FROM proj WHERE loc = valor"

Processo de Fragmentação Vertical

- **Exemplificando:**

  A1: pno A2: nome A3: orcamen A4: loc

- **Esses valores ainda não são suficientes pra formar a base de divisão**
- **Esses valores não representam o peso das frequências dos aplicativos**
- **A medida de frequência pode ser incluída na definição da medida de afinidade de atributos aff(Ai, Aj), que mede a ligação entre dois atributos de uma relação de acordo como eles são acessados pelos aplicativos**
- **Matriz de uso de atributos:**

	A_1	A_2	A_3	A_4
q_1	1	0	1	0
q_2	0	1	1	0
q_3	0	1	0	1
q_4	0	0	1	1

Processo de Fragmentação Vertical

- **Deve-se saber o peso das frequências dos aplicativos. Para isso o conceito de Afinidade entre atributos é definido.**
- **A medida de afinidade de atributos entre dois tributos Ai e Aj de uma relação R(A1, A2, ..., An) com respeito ao conj. de aplicativos Q = {q1, q2, ..., qm} é definida em função de:**
  - *$ref_l(q_k)$: número de acessos dos atributos (A_i, A_j) a cada execução de q_k no site S_l*
  - *$acc_l(q_k)$: medida de frequência de acesso do aplicativo (incluindo frequências em diferentes sites)*

Processo de Fragmentação Vertical

- **Supondo $ref_l(q_k) = 1$ pra todo k e pra todo l e as frequências:**
  - $acc_1(q_1) = 15$
  - $acc_2(q_1) = 20$
  - $acc_3(q_1) = 10$
  - $acc_1(q_2) = 5$
  - $acc_2(q_2) = 0$
  - $acc_3(q_2) = 0$
  - $acc_1(q_3) = 25$
  - $acc_2(q_3) = 25$
  - $acc_3(q_3) = 25$
  - $acc_1(q_4) = 3$
  - $acc_2(q_4) = 0$
  - $acc_3(q_4) = 0$

Processo de Fragmentação Vertical

Assim, a afinidade entre A_1 e A_3, por exemplo é dada por:

$$aff(A_1,A_3) = \sum_{k=1}^{1} \sum_{l=1}^{3} acc_l(q_k) = acc_1(q_1) + acc_2(q_1) + acc_3(q_1) = 45$$

Matriz de Afinidade dos atributos:

	A_1	A_2	A_3	A_4
A_1	45	0	45	0
A_2	0	80	5	75
A_3	45	5	53	3
A_4	0	75	3	78

- **Esta matriz é usada como base para o agrupamento em clusters de atributos com maior afinidade para posterior fragmentação**

Processo de Fragmentação Vertical

- **O processo de fragmentação envolve primeiro o agrupamento em clusters dos atributos que têm uma grande afinidade entre si, seguido pela divisão da relação de forma adequada**

Processo de Fragmentação Vertical

- **Algoritmos de agrupamento em clusters**
  - Tarefa fundamental da Frag.Vert.: agrupar atributos de uma relação em clusters.
  - Agrupar os atributos com base na matriz de afinidades.
    - Algoritmo de Energia de Ligação foi considerado para esse propósito:
      1. Atributos de mais afinidades em um cluster, atributos de menos afinidades em outro;
      2. Os agrupamentos não dependem da ordem que os itens são apresentados ao algoritmo;
      3. Complexidade de $O(n^2)$, sendo n o número de atributos;
      4. Os iter-relacionamentos secundários entre grupos de atributos reunidos em clusters são identificáveis.

Processo de Fragmentação Vertical

- O algoritmo de energia de ligação aceita como entrada a matriz de afinidade de atributos, permuta suas linhas e colunas e gera uma matriz de afinidade de agrupada em clusters
- A permutação entre linhas e colunas é feita de modo a maximizar a medida de afinidade global
- A medida de afinidade global é dada em função das medidas de afinidade entre cada para de atributos

Processo de Fragmentação Vertical

- O objetivo da atividade de divisão é encontrar conjuntos de atributos que são acessados unicamente, ou em sua maioria, por conj. de aplicativos distintos

Fragmentação híbrida

- **Consiste em realizar uma fragmentação vertical seguida de uma fragmentação horizontal ou vice-versa**
- **Ex: um conjunto de fragmentos horizontais, onde cada um deles é particionado em dois fragmentos verticais**
- **O número de níveis de aninhamento pode ser grande, mas certamente é finito**
  - No caso da fragmentação horizontal, deve-se parar quando cada fragmento consistir em uma única tupla
  - O ponto de término de uma fragmentação vertical é um atributo por fragmento

Fragmentação híbrida

- **Na prática não é possível executar muitas fragmentações verticais antes que o custo das junções se torne muito alto, uma vez que as relações globais normalizadas já possuem graus pequenos**
- **As regras de correção e as condições para a fragmentação híbrida decorrem naturalmente das regras para as fragmenta'ões horizontal e vertical**

O Problema da alocação

- Descrição do problema
- Dados
  - F = {F1 , F2 , …, Fn } fragmentos
  - S ={S1 , S2 , …, Sm } nós da rede
  - Q = {q1 , q2 ,…, qq } aplicativos
- Encontre a distribuição “ótima” de F em S

O Problema da alocação

- O caráter ótimo pode ser definido com relação a duas medidas:
- Custo mínimo
  - Comunicação + armazenamento de cada Fi em um site Sj + consultar Fi em Sj + custo de atualização de Fi em todos os sites onde ele está armazenado
- Desempenho
  - A estratégia de alocação é projetada para manter uma métrica de desempenho. Duas estratégias bem conhecidas são minimizar o tempo de resposta e maximizar o throughput em cada site.

Problema de alocação – requisitos de informação

- Informações do banco de dados
  - Seletividade dos fragmentos (quant. De tuplas que precisam ser processadas para responder uma dada consulta)
  - Tamanho dos fragmentos
- Informações de aplicativos
  - Tipos e números dos acessos (leitura e atualização)
  - Localidade dos acessos
  - O tempo de resposta máximo permitido em cada aplicativo

Problema de alocação – requisitos de informação

- Informações sobre os sites
  - Para cada site é preciso conhecer seu espaço de armazenamento e sua capacidade de processamento
  - Custo unitário de armazenamento de um dado em um site
  - Custo unitário de processamento em um site
- Informações da rede
  - Largura de banda
  - Overhead de comunicação

Modelo de alocação

- Forma Geral
  - Min(custoTotal)
- Sujeito às restrições
  - De tempo de resposta
  - De armazenamento
  - De processamento

Modelo de alocação

- **A função de custo total tem dois componentes: procesamento de consultas e armazenamento**
  - O custo de armazenamento: simples de especificar, sendo dado pelo custo total de armazenamento em todos os sites para todos os fragmentos
  - O custo de processamento de consultas: mais díficil de especificar

Modelo de alocação

- **Deve-se procurar por um esquema de alocação que, por exemplo, responda às consultas do usuário em tempo mínimo, enquanto mantém mínimo o custo de processamento.**
- **Tais modelos são dfíceis de serem desenvlovidos!**

Modelo de alocação

- **Problema de alocação de arquivos (PAA) x Problema de alocação de bancos de dados (PABD)**
- **Fragmentos não são arquivos individuais**
  - Relacionamentos têm que ser mantidos
- **Acesso aos BD é mais complicado**
  - Modelo de acesso a arquivos remotos não é aplicável
  - Relacionamento entre alocação e processamento de consultas
- **Custos adicionais que devem ser considerados**
  - Manutenção da integridade
  - Controle de concorrência